ICEIS 2016

18th International Conference on Enterprise Information Systems

PROCEEDINGS | Volume 1

Rome, Italy

25 - 28 April, 2016

www.iceis.org

SPONSORED BY: INSTICC

LOGISTICS PARTNER: SCITEVENTS SCIENCE AND TECHNOLOGY EVENTS

PAPERS AVAILABLE AT: SCITEPRESS DIGITAL LIBRARY

ICEIS 2016

Proceedings of the 18th International Conference on Enterprise Information Systems

Volume 1

Rome - Italy

April 25 - 28, 2016

Sponsored by
INSTICC - Institute for Systems and Technologies of Information, Control and Communication

Technically Co-sponsored by
IEEE SMC TC on EIS

In Cooperation with
IRC - Informatics Research Center
SWIM - IEICE Special Interest Group on Software Interprise Modelling
AEPIA - Spanish Association of Artificial Intelligence
AAAI - Association for the Advancement of Artificial Intelligence
ACM SIGMIS - ACM Special Interest Group on Management Information Systems
ACM SIGCHI - ACM Special Interest Group on Computer Human Interaction
ACM SIGAI - ACM Special Interest Group on Artificial Intelligence

Edited by Slimane Hammoudi, Leszek Maciaszek, Michele M. Missikoff, Olivier Camp and José Cordeiro

Printed in Portugal
ISBN: 978-989-758-187-8
Depósito Legal: 406875/16

http://www.iceis.org
iceis.secretariat@insticc.org

BRIEF CONTENTS

INVITED SPEAKERS

Claudia Loebbecke
University of Cologne
Germany

Sergio Gusmeroli
TXT e-solutions SPA
Italy

Wil Van Der Aalst
Technische Universiteit Eindhoven
Netherlands

Jan Vom Brocke
University of Liechtenstein
Liechtenstein

ORGANIZING COMMITTEES

CONFERENCE CO-CHAIRS

Olivier Camp, MODESTE/ESEO, France
José Cordeiro, Polytechnic Institute of Setúbal / INSTICC, Portugal

PROGRAM CO-CHAIRS

Slimane Hammoudi, ESEO, MODESTE, France
Leszek Maciaszek, Wroclaw University of Economics, Poland and Macquarie University, Sydney, Australia
Michele M. Missikoff, Institute of Sciences and Technologies of Cognition, ISTC-CNR, Italy

SECRETARIAT

Vitor Pedrosa, INSTICC, Portugal

GRAPHICS PRODUCTION AND WEBDESIGNER

André Lista, INSTICC, Portugal
Mara Silva, INSTICC, Portugal

WEBMASTER

Susana Rodrigues, INSTICC, Portugal

PROGRAM COMMITTEE

Miguel Angel Martinez Aguilar, University of Murcia, Spain

Adeel Ahmad, Laboratoire d'Informatique Signal et Image de la Côte d'Opale, France

Patrick Albers, ESEO - Ecole Superieure D'Electronique de L'Ouest, France

Mohammad Al-Shamri, Ibb University, Yemen

Rainer Alt, University of Leipzig, Germany

Andreas S. Andreou, Cyprus University of Technology, Cyprus

Oscar Avila, Universidad de los Andes, Colombia

Tamara Babaian, Bentley University, United States

Cecilia Baranauskas, State University of Campinas - Unicamp, Brazil

Ken Barker, University of Calgary, Canada

Jean-Paul Barthès, Heudiasyc, JRU CNRS 7253, Université de Technologie de Compiègne, France

Lamia Hadrich Belguith, ANLP Research Group, MIRACL, University of Sfax, Tunisia

Orlando Belo, University of Minho, Portugal

Domenico Beneventano, Universita' di Modena e Reggio Emilia, Italy

Jorge Bernardino, Polytechnic Institute of Coimbra - ISEC, Portugal

Frederique Biennier, INSA Lyon, France

Sandro Bimonte, Irstea, France

Marko Bohanec, Jožef Stefan Institute, Slovenia

Jean-Louis Boulanger, CERTIFER, France

Daniel Antonio Callegari, PUC-RS Pontificia Universidade Catolica do Rio Grande do Sul, Brazil

Luis M. Camarinha-Matos, New University of Lisbon, Portugal

Roy Campbell, University of Illinoiis at Urbana-Champaign, United States

Manuel Isidoro Capel-Tuñón, University of Granada, Spain

Glauco Carneiro, Salvador University (UNIFACS), Brazil

Angélica Caro, University of Bio-Bio, Chile

Nunzio Casalino, LUISS Guido Carli University, Italy

Marco A. Casanova, PUC-Rio, Brazil

Luca Cernuzzi, Universidad Católica "Nuestra Señora de la Asunción", Paraguay

Shiping Chen, CSIRO, Australia

Max Chevalier, Institut de Recherche en Informatique de Toulouse UMR 5505, France

Nan-Hsing Chiu, Chien Hsin University of Science and Technology, Taiwan

Witold Chmielarz, Warsaw University, Poland

Daniela Barreiro Claro, Universidade Federal da Bahia (UFBA), Brazil

Pedro Henrique Gouvêa Coelho, State University of Rio de Janeiro, Brazil

Francesco Colace, Università Degli Studi di Salerno, Italy

Cesar Collazos, Universidad del Cauca, Colombia

Antonio Corral, University of Almeria, Spain

Mariela Cortés, State University of Ceará, Brazil

Sharon Cox, Birmingham City University, United Kingdom

Broderick Crawford, Pontificia Universidad Catolica de Valparaiso, Chile

Vincenzo Deufemia, Università di Salerno, Italy

Dulce Domingos, Faculty of Science - University of Lisbon, Portugal

César Domínguez, Universidad de La Rioja, Spain

Sophie Ebersold, Université Toulouse II-Le Mirail, France

Hans-Dieter Ehrich, Technische Universitaet Braunschweig, Germany

Fabrício Enembreck, Pontifical Catholic University of Paraná, Brazil

Sean Eom, Southeast Missouri State University, United States

João Faria, FEUP - Faculty of Engineering of the University of Porto, Portugal

Antonio Fariña, University of A Coruña, Spain

Edilson Ferneda, Catholic University of Brasília, Brazil

Maria João Silva Costa Ferreira, Universidade Portucalense, Portugal

Paulo Ferreira, INESC-ID / IST, Portugal

George Feuerlicht, University of Technology, Sydney (UTS), Australia

António Figueiredo, University of Coimbra, Portugal

Rita Francese, Università degli Studi di Salerno, Italy

Ana Fred, Instituto de Telecomunicações / IST, Portugal

Lixin Fu, University of North Carolina, Greensboro, United States

Johannes Gettinger, University of Hohenheim, Germany

George Giaglis, Athens University of Economics and Business, Greece

Daniela Giordano, University of Catania, Italy

Feliz Gouveia, University Fernando Pessoa / Cerem, Portugal

Virginie Govaere, INRS, France

Janis Grabis, Riga Technical University, Latvia

Maria Carmen Penadés Gramaje, Universitat Politècnica de València, Spain

Sven Groppe, University of Lübeck, Germany

Wieslawa Gryncewicz, Wroclaw University of Economics, Poland

Rune Gustavsson, Blekinge Institute of Technology, Sweden

Hatim HAFIDDI, INPT, Morocco

Karin Harbusch, Universität Koblenz-Landau, Germany

Wladyslaw Homenda, Warsaw University of Technology, Poland

Wei-Chiang Hong, Nanjing Tech University, China, Taiwan

Miguel J. Hornos, University of Granada, Spain

Kai-I Huang, Tunghai University, Taiwan

Miroslav Hudec, University of Economics in Bratislava, Slovak Republic

San-Yih Hwang, National Sun Yat-sen University, Taiwan

Abdessamad Imine, Laboratoire Lorrain de Recherche en Informatique et Ses Applications, France

Arturo Jaime, Universidad de La Rioja, Spain

Kai Jakobs, RWTH Aachen University, Germany

Paul Johannesson, Royal Institute of Technology, Sweden

Nikitas Karanikolas, Technological Educational Institute of Athens (TEI-A), Greece

Dimitrios Katsaros, University of Thessaly, Greece

Andrea Kienle, University of Applied Sciences, Dortmund, Germany

Marite Kirikova, Riga Technical University, Latvia

Alexander Knapp, Universität Augsburg, Germany

Natallia Kokash, Leiden University, Netherlands

Fotis Kokkoras, TEI of Thessaly, Greece

Christophe Kolski, University of Valenciennes, France

John Krogstie, NTNU, Norway

Rob Kusters, Eindhoven University of Technology & Open University of the Netherlands, Netherlands

Wim Laurier, Université Saint-Louis, Belgium

Ramon Lawrence, University of British Columbia Okanagan, Canada

Jintae Lee, Leeds School of Business at University of Colorado, Boulder, United States

Alain Leger, France Telecom Orange Labs, France

Joerg Leukel, University of Hohenheim, Germany

Lei Li, Hefei University of Technology, China

Da-Yin Liao, Applied Wireless Identifications, United States

Luis Jiménez Linares, University of de Castilla-La Mancha, Spain

Panagiotis Linos, Butler University, United States

Stephane Loiseau, LERIA, University of Angers, France

João Correia Lopes, INESC TEC, Faculdade de Engenharia da Universidade do Porto, Portugal

Maria Filomena Cerqueira de Castro Lopes, Universidade Portucalense Infante D. Henrique, Portugal

Wendy Lucas, Bentley University, United States

André Ludwig, University of Leipzig, Germany

Mark Lycett, Brunel University, United Kingdom

Jose Antonio Macedo, Federal University of Ceara, Brazil

Leszek Maciaszek, Wroclaw University of Economics, Poland and Macquarie University, Sydney, Australia

Cristiano Maciel, Universidade Federal de Mato Grosso, Brazil

Rita Suzana Pitangueira Maciel, Federal University of Bahia, Brazil

Riccardo Martoglia, University of Modena and Reggio Emilia, Italy

Katsuhisa Maruyama, Ritsumeikan University, Japan

David Martins de Matos, L2F / INESC-ID Lisboa / Instituto Superior Técnico, Portugal

Wolfgang Mayer, University of South Australia, Australia

Brad Mehlenbacher, North Carolina State University, United States

Jerzy Michnik, University of Economics in Katowice, Poland

Marek Milosz, Lublin University of Technology, Poland

Michele Missikoff, ISTC-CNR, Italy

Pascal Molli, LINA, University of Nantes, France

Lars Mönch, FernUniversität in Hagen, Germany

Francisco Montero, University of Castilla-la Mancha, Spain

Carlos León de Mora, University of Seville, Spain

João Luís Cardoso de Moraes, Federal University of São Carlos, Brazil

Fernando Moreira, Universidade Portucalense, Portugal

Pietro Murano, Oslo and Akershus University College of Applied Sciences, Norway

Tomoharu Nakashima, Osaka Prefecture University, Japan

Alvaro Navas, Universidad Politécnica de Madrid, Spain

Rabia Nessah, IESEG School of Management, France

Vincent Ng, The Hong Kong Polytechnic University, Hong Kong

Ovidiu Noran, Griffith University, Australia

Edson OliveiraJr, State University of Maringá, Brazil

Andrés Muñoz Ortega, Catholic University of Murcia (UCAM), Spain

Claus Pahl, Dublin City University, Ireland

Philippe Palanque, Institut de Recherche en Informatique de Toulouse, France

Tadeusz Pankowski, Poznan University of Technology, Poland

Hugo Parada, UPM, Spain

Eric Pardede, La Trobe University, Australia

Viviana Patti, University of Torino, Italy

Luis Ferreira Pires, University of Twente, Netherlands

Pierluigi Plebani, Politecnico di Milano, Italy

Geert Poels, Ghent University, Belgium

Luigi Pontieri, National Research Council (CNR), Italy

Filipe Portela, Centro Algoritmi, Universidade do Minho, Portugal

Robin Qiu, Pennsylvania State University, United States

Ricardo J. Rabelo, Federal University of Santa Catarina, Brazil

Daniele Radicioni, University of Turin, Italy

T. Ramayah, Universiti Sains Malaysia, Malaysia

Pedro Ramos, Instituto Superior das Ciências do Trabalho e da Empresa, Portugal

Francisco Regateiro, Instituto Superior Técnico, Portugal

Ulrich Reimer, University of Applied Sciences St. Gallen, Switzerland

Nuno de Magalhães Ribeiro, Universidade Fernando Pessoa, Portugal

Michele Risi, University of Salerno, Italy

Sérgio Assis Rodrigues, COPPE/UFRJ – Federal University of Rio de Janeiro, Brazil

Alfonso Rodriguez, University of Bio-Bio, Chile

Daniel Rodriguez, University of Alcalá, Spain

Oscar Mario Rodriguez-Elias, Institute of Technology of Hermosillo, Mexico

Luciana Alvim Santos Romani, Embrapa Agricultural Informatics, Brazil

Jose Raul Romero, University of Cordoba, Spain

David G. Rosado, University of Castilla-la Mancha, Spain

Michael Rosemann, Queensland University of Technology, Australia

Gustavo Rossi, Lifia, Argentina

Francisco Ruiz, Universidad de Castilla-La Mancha, Spain

Indrajit Saha, National Institute of Technical Teachers' Training & Research, India

Belen Vela Sanchez, Rey Juan Carlos University, Spain

Luis Enrique Sánchez, Universidad de Castilla-la Mancha, Spain

Manuel Filipe Santos, University of Minho, Portugal

Sissel Guttormsen Schär, Institute for Medical Education, Switzerland

Isabel Seruca, Universidade Portucalense, Portugal

Ahm Shamsuzzoha, Sultan Qaboos University, Oman

Jianhua Shao, Cardiff University, United Kingdom

Markus Siepermann, TU Dortmund, Germany

Alberto Rodrigues Silva, Instituto Superior Técnico, Portugal

Sean Siqueira, Federal University of the State of Rio de Janeiro (UNIRIO), Brazil

Spiros Sirmakessis, Technological Educational Institution of Messolongi, Greece

Hala Skaf-molli, Nantes University, France

Michel Soares, Federal University of Sergipe, Brazil

Ricardo Soto, Pontificia Universidad Catolica de Valparaiso, Chile

Chantal Soule-Dupuy, Universite Toulouse 1, France

Patricia Souza, UFMT, Brazil

Hatem Ben Sta, Tunisia University, Tunisia

Clare Stanier, Stafordshire University, United Kingdom

Chris Stary, Johannes Kepler University of Linz, Austria

Vijayan Sugumaran, Oakland University, United States

Hiroki Suguri, Miyagi University, Japan

Lily Sun, University of Reading, United Kingdom

Jerzy Surma, Warsaw School of Economics, Poland

Ryszard Tadeusiewicz, AGH University of Science and Technology, Poland

Tania Tait, Maringá State University, Brazil

Mohan Tanniru, Oakland University, United States

Sotirios Terzis, University of Strathclyde, United Kingdom

Claudine Toffolon, Université du Maine, France

José Tribolet, INESC-ID/Instituto Superior Técnico, Portugal

Theodoros Tzouramanis, University of the Aegean, Greece

Domenico Ursino, Università degli Studi Mediterranea Reggio Calabria, Italy

Vadim Vagin, Moscow Power Engineering Institute (National Research University), Russian Federation

José Ângelo Braga de Vasconcelos, Universidade Atlântica, Portugal

Michael Vassilakopoulos, University of Thessaly, Greece

Maria Esther Vidal, Universidad Simon Bolivar, Venezuela

Stephanie Vie, University of Central Florida, United States

Gualtiero Volpe, Università degli Studi di Genova, Italy

Bing Wang, University of Hull, United Kingdom

Dariusz Wawrzyniak, Wroclaw University of Economics, Poland

Hans Weghorn, BW Cooperative State University Stuttgart, Germany

Hans Weigand, Tilburg University, Netherlands

Robert Wrembel, Poznan University of Technology, Poland

Stanislaw Wrycza, University of Gdansk, Poland

Mudasser Wyne, National University, United States

Hongji Yang, Bath Spa University, United Kingdom

Stefano Za, CeRSI - LUISS Guido Carli University, Italy

Eugenio Zimeo, University of Sannio, Italy

AUXILIARY REVIEWERS

Boris Almonacid, Pontificia Universidad Católica de Valparaíso, Chile

Marcio Bera, State University of Maringá, Brazil

Solvita Berzisa, Riga Technical university, Latvia

Magdalena Cantabella, Universidad Católica San Antonio de Murcia, Spain

Claudia Cappelli, UNIRIO - Universidade Federal do Estado do Rio de Janeiro, Brazil

Luisa Carpente, University of A Coruña, Spain

Fernando William Cruz, Universidade de Brasilia, Brazil

Delia Irazù Hernandez Farias, UPV - UNITO, Spain

Fausto Fasano, University of Molise, Italy

Cristian Galleguillos, Pontificia Universidad Católica de Valparaíso, Chile

Javier David Fernández García, Vienna University of Economics and Business, Austria

Ricardo Geraldi, State University of Maringá, Brazil

Rafael Glauber, Federal University of Bahia, Brazil

Magalí González, Universidad Católica Nuestra Señora de la Asunción, Paraguay

Djilali Idoughi, University A. Mira of Bejaia, Algeria

Christos Kalyvas, University of the Aegean, Greece

Janis Kampars, Riga Technical University, Latvia

Shixong Liu, University of Reading, United Kingdom

Leandros Maglaras, De Montfort University, United Kingdom

Anderson Marcolino, University of São Paulo, Brazil

Eirini Molla, University of the Aegean, Greece

Dario Di Nucci, University of Salerno, Italy

Fabio Palomba, University of Salerno, Italy

Roberto Pereira, University of Campinas (UNICAMP), Brazil

Victor Reyes, Pontifica Universidad Catolica de Valparaiso, Chile

Jorge Saldivar, Catholic University "Nuestra Señora de la Asunción", Paraguay

SELECTED PAPERS BOOK

A number of selected papers presented at ICEIS 2016 will be published by Springer in a LNBIP Series book. This selection will be done by the Conference Co-chairs and Program Co-chairs, among the papers actually presented at the conference, based on a rigorous review by the ICEIS 2016 Program Committee members.

FOREWORD

This book contains the proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016), which was sponsored by the Institute for Systems and Technologies of Information, Control and Communication (INSTICC), held in cooperation with the Association for the Advancement of Artificial Intelligence (AAAI), IEICE Special Interest Group on Software Interprise Modelling (SWIM), ACM SIGMIS - ACM Special Interest Group on Management Information Systems, ACM SIGAI - ACM Special Interest Group on Artificial Intelligence, ACM SIGCHI - ACM Special Interest Group on Computer Human Interaction, the Spanish Association for Artificial Intelligence (AEPIA), the Informatics Research Center (IRC) and technically co-sponsored by IEEE SMC (Systems, Men, and Cybernetics) Technical Committee on Enterprise Information Systems (TCEIS). This year ICEIS was held in Rome, Italy from 25 - 28 April, 2016.

The purpose of the 18th International Conference on Enterprise Information Systems is to bring together researchers, engineers and practitioners from the areas of "Databases and Information Systems Integration", "Artificial Intelligence and Decision Support Systems", "Information Systems Analysis and Specification", "Software Agents and Internet Computing", "Human-Computer Interaction" and "Enterprise Architecture", interested in the advances and business applications of information systems.

ICEIS 2016 received 257 paper submissions from 42 countries in all continents, which makes it one of the largest conferences in the World in the area of Information Systems, thus demonstrating the success and global dimension of this conference. From these, 42 papers were selected for publication and presentation at the Conference as full papers. These numbers, leading to a full-paper acceptance ratio of 16%, show the intention of preserving a high quality forum for this conference, a quality that we intend to maintain in the future, for the next editions of this conference.

The high number and high quality of the received papers imposed difficult choices in the selection process. To evaluate each submission, a double blind paper review was performed by the Program Committee, whose members are highly qualified researchers in ICEIS topic areas.

All presented papers will be available at the SCITEPRESS Digital Library and will be submitted for indexation by Thomson Reuters Conference Proceedings Citation Index (ISI), INSPEC, DBLP, EI (Elsevier Index) and Scopus.

Additionally, a short list of presented papers will be selected to be expanded into a forthcoming book of ICEIS 2016 Selected Papers to be published by Springer in the LNBIP Series.

The technical program of the conference included a panel and 4 invited talks delivered by internationally distinguished speakers, namely: Claudia Loebbecke (University of Cologne, Germany), Sergio Gusmeroli (TXT e-solutions SPA, Italy), Wil Van Der Aalst (Technische Universiteit Eindhoven, Netherlands) and Jan Vom Brocke (University of Liechtenstein, Liechtenstein). Their participation positively contributes to reinforce the overall quality of the Conference and to provide a deeper understanding of the fields addressed by the conference.

Moreover, ICEIS 2016 had a Doctoral Consortium on Enterprise Information Systems and 1 tutorial. We are thankful to the Conference Co-chairs (Olivier Camp and José Cordeiro) and Program Co-chairs (Slimane Hammoudi, Leszek Maciaszek and Michele M. Missikoff) for their dedication and hard work in organizing these events.

We sincerely thank all the authors for their submissions and participation in ICEIS 2016. Furthermore, we would like to thank all the members of the program committee and reviewers, who helped us with their expertise, dedication and time. We would also like to thank the invited speakers for their excellent contribution in sharing their knowledge and vision and the workshop/special session chairs whose collaboration with ICEIS 2015 was much appreciated. Finally, we gratefully acknowledge the professional support of the ICEIS 2016 team for all organizational processes.

We hope that all colleagues find this a fruitful and inspiring conference. We hope to contribute to the development of the Enterprise Information Systems community and look forward to having additional research results presented at the next edition of ICEIS, details of which are available at http://www.iceis.org.

Slimane Hammoudi
ESEO, MODESTE, France

Leszek Maciaszek
Wroclaw University of Economics, Poland and Macquarie University, Sydney, Australia

Michele M. Missikoff
Institute of Sciences and Technologies of Cognition, ISTC-CNR, Italy

Olivier Camp
MODESTE/ESEO, France

José Cordeiro
Polytechnic Institute of Setúbal / INSTICC, Portugal

CONTENTS

SHORT PAPERS

INVITED SPEAKERS

KEYNOTE SPEAKERS

Big Data Analytics
Just More or Conceptually Different?

Claudia Loebbecke
University of Cologne, Germany

Abstract: In the era of so-called big data, analytics for designing and delivering innovative services and actionable insights goes beyond dealing faster and smarter with more data. Done well, harnessing big data analytics will drive fundamentally transformed approaches to value creation – in business, industry sectors, society, higher education, and research.

This presentation will outline how big data analytics can empower organizations in the big data era and hopefully open the discussion on proactively shaping of new opportunities.

BRIEF BIOGRAPHY

Claudia Loebbecke is a professor of Business, Media and Technology Management and Director of the Department of Media and Technology Management at the University of Cologne, Germany. 2005–2006 she was elected president of the Association for Information Systems (AIS). She serves as Senior Editor of the Journal of Strategic Information Systems (JSIS), as Advisory Board Member of Information Systems Research (ISR) and of the Journal of Information Technology (JIT), and on the Editorial Board of the Information Systems Journal (ISJ) and Communications of the Association for Information Systems (CAIS). Claudia received a Masters (1990) and a Ph.D. (1995) in Business Administration, both from the University of Cologne, Germany, and an MBA from Indiana University, Bloomington, Indiana, U.S. (1991). In 2011, she co-authored the study "Assessing Cloud Readiness," which won the Research Competition of the Society for Information Management. She has published over fifty internationally peer-reviewed journal articles.

Loebbecke, C.
Big Data Analytics - Just More or Conceptually Different?.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, page 5
ISBN: 978-989-758-187-8

The Sensing Enterprise
Enterprise Information Systems in the Internet of Things

Sergio Gusmeroli
TXT e-solutions SPA, Italy

Abstract: The keynote aims at describing how recent IT innovations in the field of IoT (e.g. cyber physical systems, smart networks, edge computing, smart objects, business intelligence, data analytics) are influencing the evolution of Enterprise Information Systems. Thanks to the advent of IOT, Enterprise PLM systems are abandoning the walled garden of Design and Engineering, while embracing the whole product lifecycle, including post-sales services and addressing circular economy challenges, becoming this way "Things Lifecycle Management Systems". At the same time, MES (Manufacturing Execution Systems) need to consider Industry 4.0 evolution in production systems and the advent of Cyber Physical and Systems. What it is not fully clear up to now is the IOT-driven evolution of ERP and SCM systems and how decision making at the level of configuration, planning and scheduling of enterprises' resources could be implemented by distributed edge-computing architectures. We call this new concept "The Sensing Proactive Enterprise". The speech is inspired by several EC-funded R&I projects in the field of "IOT for Enterprise" under the FP7 and H2020 Framework Programmes in the Net Innovation unit E3 of the Future Internet (DG CNECT).

BRIEF BIOGRAPHY

Sergio Gusmeroli is a Senior Advisor for the Research and Innovation Unit in TXT e-Solutions SPA (www.txtgroup.com). As a researcher, Sergio has recently and is coordinating large scale FP7 projects and H2020 R&I actions (e.g. Elliot MSEE FITMAN OSMOSE PSYMBIOSYS) in the field of IoT technologies especially applied to Manufacturing Industries. As an innovation manager, Sergio has been co-conducting with IDC the EC-commissioned study Definition of a research and innovation policy leveraging Cloud Computing and IoT Combination, SMART 2013/0037 and will soon coordinate an H2020 Innovation Action about full adoption of IoT-enabled Cyber Physical Production Systems in Industry.

Gusmeroli, S.
The Sensing Enterprise - Enterprise Information Systems in the Internet of Things.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, page 7
ISBN: 978-989-758-187-8

Green Data Science
Using Big Data in an "Environmentally Friendly" Manner

Wil M. P. van der Aalst
Eindhoven University of Technology, Department of Mathematics and Computer Science,
PO Box 513, NL-5600 MB Eindhoven, The Netherlands
w.m.p.v.d.aalst@tue.nl

Keywords: Data Science, Big Data, Fairness, Confidentiality, Accuracy, Transparency, Process Mining.

Abstract: The widespread use of "Big Data" is heavily impacting organizations and individuals for which these data are collected. Sophisticated data science techniques aim to extract as much value from data as possible. Powerful mixtures of Big Data and analytics are rapidly changing the way we do business, socialize, conduct research, and govern society. Big Data is considered as the "new oil" and data science aims to transform this into new forms of "energy": insights, diagnostics, predictions, and automated decisions. However, the process of transforming "new oil" (data) into "new energy" (analytics) may negatively impact citizens, patients, customers, and employees. Systematic discrimination based on data, invasions of privacy, non-transparent life-changing decisions, and inaccurate conclusions illustrate that data science techniques may lead to new forms of "pollution". We use the term "Green Data Science" for technological solutions that enable individuals, organizations and society to reap the benefits from the widespread availability of data while ensuring fairness, confidentiality, accuracy, and transparency. To illustrate the scientific challenges related to "Green Data Science", we focus on process mining as a concrete example. Recent breakthroughs in process mining resulted in powerful techniques to discover the real processes, to detect deviations from normative process models, and to analyze bottlenecks and waste. Therefore, this paper poses the question: How to benefit from process mining while avoiding "pollutions" related to unfairness, undesired disclosures, inaccuracies, and non-transparency?

1 INTRODUCTION

In recent years, data science emerged as a new and important discipline. It can be viewed as an amalgamation of classical disciplines like statistics, data mining, databases, and distributed systems. We use the following definition: *"Data science is an interdisciplinary field aiming to turn data into real value. Data may be structured or unstructured, big or small, static or streaming. Value may be provided in the form of predictions, models learned from data, or any type of data visualization delivering insights. Data science includes data extraction, data preparation, data exploration, data transformation, storage and retrieval, computing infrastructures, various types of mining and learning, presentation of explanations and predictions, and the exploitation of results taking into account ethical, social, legal, and business aspects."* (Aalst, 2016).

Related to data science is the overhyped term "Big Data" that is used to refer to the massive amounts of data collected. Organizations are heavily investing in Big Data technologies, but at the same time citizens, patients, customers, and employees are concerned about the use of their data. We live in an era characterized by unprecedented opportunities to sense, store, and analyze data related to human activities in great detail and resolution. This introduces new risks and intended or unintended abuse enabled by powerful analysis techniques. Data may be sensitive and personal, and should not be revealed or used for proposes different from what was agreed upon. Moreover, analysis techniques may discriminate minorities even when attributes like gender and race are removed. Using data science technology as a "black box" making life-changing decisions (e.g., medical prioritization or mortgage approvals) triggers a variety of ethical dilemmas.

Sustainable data science is only possible when citizens, patients, customers, and employees are *protected against irresponsible uses of data* (big or small). Therefore, we need to separate the "good" and "bad" of data science. Compare this with environmentally friendly forms of green energy (e.g. solar power) that overcome problems related to traditional forms of energy. Data science may result in

Aalst, W.
Green Data Science - Using Big Data in an "Environmentally Friendly" Manner.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 9-21
ISBN: 978-989-758-187-8

unfair decision making, undesired disclosures, inaccuracies, and non-transparency. These irresponsible uses of data can be viewed as "pollution". Abandoning the systematic use of data may help to overcome these problems. However, this would be comparable to abandoning the use of energy altogether. Data science is used to make products and services more reliable, convenient, efficient, and cost effective. Moreover, most new products and services depend on the collection and use of data. Therefore, we argue that the "prohibition of data (science)" is not a viable solution.

In this paper, we coin the term *"Green Data Science"* (GDS) to refer to the collection of techniques and approaches trying to reap the benefits of data science and Big Data while ensuring fairness, confidentiality, accuracy, and transparency. *We believe that technological solutions can be used to avoid pollution and protect the environment in which data is collected and used.* Section 2 elaborates on the following four challenges:

- **Fairness** – Data Science without prejudice: How to avoid unfair conclusions even if they are true?
- **Confidentiality** – Data Science that ensures confidentiality: How to answer questions without revealing secrets?
- **Accuracy** – Data Science without guesswork: How to answer questions with a guaranteed level of accuracy?
- **Transparency** – Data Science that provides transparency: How to clarify answers such that they become indisputable?

Concerns related to privacy and personal data protection triggered legislation like the EU's Data Protection Directive. *Directive 95/46/EC* ("on the protection of individuals with regard to the processing of personal data and on the free movement of such data") of the European Parliament and the Council was adopted on 24 October 1995 (European Commission, 1995). The *General Data Protection Regulation* (GDPR) is currently under development and aims to strengthen and unify data protection for individuals within the EU (European Commission, 2015). GDPR will replace Directive 95/46/EC and is expected to be finalized in Spring 2016 and will be much more restrictive than earlier legislation. Sanctions include fines of up to 4% of the annual worldwide turnover. GDPR and other forms of legislation limiting the use of data, may prevent the use of data science also in situations where data is used in a positive manner. Prohibiting the collection and systematic use of data is like turning back the clock. Next to legislation, positive technological solutions are needed to ensure fairness, confidentiality, accuracy, and transparency. By just imposing restrictions, individuals, organizations and society cannot exploit data (science) in a positive way.

The four challenges discussed in Section 2 are quite general. Therefore, we focus on a concrete subdiscipline in data science in Section 3: *Process Mining* (Aalst, 2011). Process mining seeks the confrontation between event data (i.e., observed behavior) and process models (hand-made or discovered automatically). Event data are related to explicit process models, e.g., Petri nets or BPMN models. For example, process models are discovered from event data or event data are replayed on models to analyze compliance and performance. *Process mining provides a bridge between data-driven approaches (data mining, machine learning and business intelligence) and process-centric approaches (business process modeling, model-based analysis, and business process management/reengineering).* Process mining results may drive redesigns, show the need for new controls, trigger interventions, and enable automated decision support. Individuals *inside* (e.g., end-users and workers) and *outside* (e.g., customers, citizens, or patients) the organization may be impacted by process mining results. Therefore, Section 3 lists process mining challenges related to fairness, confidentiality, accuracy, and transparency.

In the long run, data science is only sustainable if we are willing to address the problems discussed in this paper. Rather than abandoning the use of data altogether, we should find positive technological ways to protect individuals.

2 FOUR CHALLENGES

Figure 1 sketches the "data science pipeline". Individuals interact with a range of hardware/software systems (information systems, smartphones, websites, wearables, etc.) ❶. Data related to machine and interaction events are collected ❷ and preprocessed for analysis ❸. During preprocessing data may be transformed, cleaned, anonymized, de-identified, etc. Models may be learned from data or made/modified by hand ❹. For compliance checking, models are often normative and made by hand rather than discovered from data. Analysis results based on data (and possibly also models) are presented to analysts, managers, etc. ❺ or used to influence the behavior of information systems and devices ❻. Based on the data, decisions are made or recommendations are provided. Analysis results may also be used to change systems, laws, procedures, guidelines, responsibilities, etc. ❼.

Figure 1 also lists the four challenges discussed

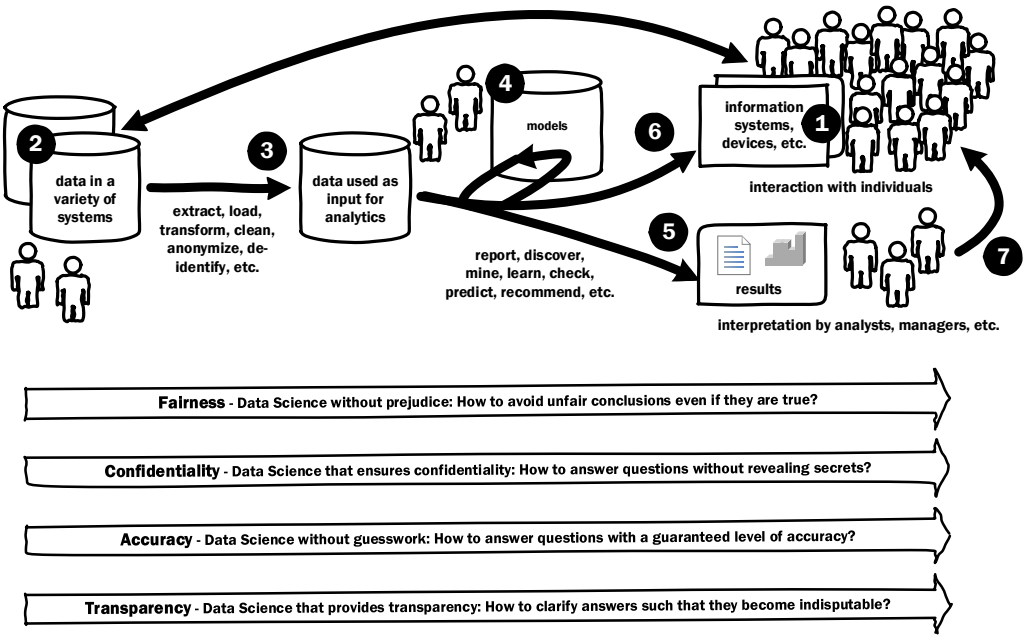


Figure 1: The "data science pipeline" facing four challenges.

in the remainder of this section. Each of the challenges requires an understanding of the whole data pipeline. Flawed analysis results or bad decisions may be caused by different factors such as a sampling bias, careless preprocessing, inadequate analysis, or an opinionated presentation.

2.1 Fairness - Data Science without Prejudice: How to Avoid Unfair Conclusions Even if they are True?

Data science techniques need to ensure *fairness*: Automated decisions and insights should not be used to discriminate in ways that are unacceptable from a legal or ethical point of view. Discrimination can be defined as "the harmful treatment of an individual based on their membership of a specific group or category (race, gender, nationality, disability, marital status, or age)". However, most analysis techniques *aim to discriminate* among groups. Banks handing out loans and credit cards try to discriminate between groups that will pay their debts and groups that will run into financial problems. Insurance companies try to discriminate between groups that are likely to claim and groups that are less likely to claim insurance. Hospitals try to discriminate between groups for which a particular treatment is likely to be effective and groups for which this is less likely. Hiring employees, providing scholarships, screening suspects, etc. can all be seen as classification problems: The goal is to explain a response variable (e.g., person will pay back the loan) in terms of predictor variables (e.g., credit history, employment status, age, etc.). Ideally, the learned model explains the response variable as good as possible without discriminating on the basis of sensitive attributes (race, gender, etc.).

To explain *discrimination discovery* and *discrimination prevention*, let us consider the set of all (potential) customers of some insurance company specializing in car insurance. For each customer we have the following variables:

- name,
- birthdate,
- gender (male or female),
- nationality,
- car brand (Alfa, BMW, etc.),
- years of driving experience,
- number of claims in the last year,
- number of claims in the last five years, and
- status (insured, refused, or left).

The status field is used to distinguish current customers (status=insured) from customers that were refused (status=refused) or that left the insurance company during the last year (status=left). Customers that were refused or that left more than a year ago are removed from the data set.

Techniques for *discrimination discovery* aim to identify groups that are discriminated based on *sensitive* variables, i.e., variables that should not matter.

For example, we may find that "males have a higher likelihood to be rejected than females" or that "foreigners driving a BMW have a higher likelihood to be rejected than Dutch BMW drivers". Discrimination may be caused by human judgment or by automated decision algorithms using a predictive model. The decision algorithms may discriminate due to a sampling bias, incomplete data, or incorrect labels. If earlier rejections are used to learn new rejections, then prejudices may be reinforced. Similar "self-fulfilling prophecies" can be caused by sampling or missing values.

Even when there is no intent to discriminate, discrimination may still occur. Even when the automated decision algorithm does not use gender and uses only non-sensitive variables, the actual decisions may still be such that (fe)males or foreigners have a much higher probability to be rejected. The decision algorithm may also favor more frequent values for a variable. As a result, minority groups may be treated unfairly.

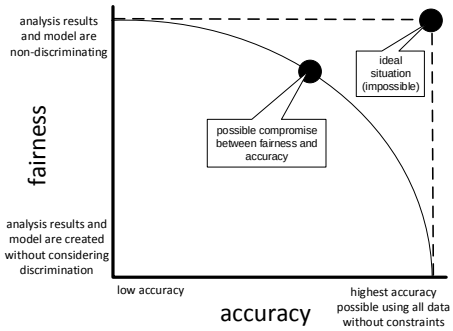


Figure 2: Tradeoff between fairness and accuracy.

Discrimination prevention aims to create automated decision algorithms that do not discriminate using sensitive variables. It is not sufficient to remove these sensitive variables: Due to correlations and the handling of outliers, unintentional discrimination may still take place. One can add constraints to the decision algorithm to ensure fairness using a predefined criterion. For example, the constraint "males and females should have approximately the same probability to be rejected" can be added to a decision-tree learning algorithm. Next to adding algorithm-specific constraints used during analysis one can also use preprocessing (modify the input data by resampling or relabeling) or postprocessing (modify models, e.g., relabel mixed leaf nodes in a decision tree). In general there is often a *trade-off between maximizing accuracy and minimizing discrimination* (see Figure 2). By rejecting fewer males (better fairness), the insurance company may need to pay more claims.

Discrimination prevention often needs to use sensitive variables (gender, age, nationality, etc.) to ensure fairness. This creates a *paradox*, e.g., information on gender needs to be used to avoid discrimination based on gender.

The first paper on discrimination-aware data mining appeared in 2008 (Pedreshi et al., 2008). Since then, several papers mostly focusing on fair classification appeared: (Calders and Verwer, 2010; Kamiran et al., 2010; Ruggieri et al., 2010). These examples show that unfairness during analysis can be actively prevented. However, unfairness is not limited to classification and more advanced forms of analytics also need to ensure fairness.

2.2 Confidentiality - Data Science that Ensures Confidentiality: How to Answer Questions without Revealing Secrets?

The application of data science techniques should not reveal certain types of personal or otherwise sensitive information. Often personal data need to be kept *confidential*. The General Data Protection Regulation (GDPR) currently under development (European Commission, 2015) focuses on personal information:

"The principles of data protection should apply to any information concerning an identified or identifiable natural person. Data including pseudonymized data, which could be attributed to a natural person by the use of additional information, should be considered as information on an identifiable natural person. To determine whether a person is identifiable, account should be taken of all the means reasonably likely to be used either by the controller or by any other person to identify the individual directly or indirectly. To ascertain whether means are reasonably likely to be used to identify the individual, account should be taken of all objective factors, such as the costs of and the amount of time required for identification, taking into consideration both available technology at the time of the processing and technological development. The principles of data protection should therefore not apply to anonymous information, that is information which does not relate to an identified or identifiable natural person or to data rendered anonymous in such a way that the data subject is not or no longer identifiable."

Confidentiality is not limited to personal data. Companies may want to hide sales volumes or production times when presenting results to certain stakeholders. One also needs to bear in mind that few information systems hold information that can be shared or analyzed without limits (e.g., the existence of personal data cannot be avoided). The "data science pipeline" depicted in Figure 1 shows that there are dif-

ferent types of data having different audiences. Here we focus on: (1) the "raw data" stored in the information system ❷, (2) the data used as input for analysis ❸, and (3) the analysis results interpreted by analysts and managers ❺. Whereas the raw data may refer to individuals, the data used for analysis is often (partly) de-identified, and analysis results may refer to aggregate data only. It is important to note that confidentiality may be endangered along the whole pipeline and includes analysis results.

Consider a data set that contains sensitive information. Records in such a data set may have three types of variables:

- *Direct Identifiers:* Variables that uniquely identify a person, house, car, company, or other entity. For example, a social security number identifies a person.
- *Key Variables:* Subsets of variables that together can be used to identify some entity. For example, it may be possible to identify a person based on gender, age, and employer. A car may be uniquely identified based on registration date, model, and color. Key variables are also referred to as *implicit identifiers* or *quasi identifiers*.
- *Non-identifying Variables:* Variables that cannot be used to identify some entity (direct or indirect).

Confidentiality is impaired by unintended or malicious disclosures. We consider three types of such disclosures:

- *Identity Disclosure:* Information about an entity (person, house, etc.) is revealed. This can be done through direct or implicit identifiers. For example, the salaries of employees are disclosed unintentionally or an intruder is able to retrieve patient data.
- *Attribute Disclosure:* Information about an entity can be derived indirectly. If there is only one male surgeon in the age group 40-45, then aggregate data for this category reveals information about this person.
- *Partial Disclosure:* Information about a group of entities can be inferred. Aggregate information on male surgeons in the age group 40-45 may disclose an unusual number of medical errors. These cannot be linked to a particular surgeon. Nevertheless, one may conclude that surgeons in this group are more likely to make errors.

De-identification of data refers to the process of removing or obscuring variables with the goal to minimize unintended disclosures. In many cases *re-identification* is possible by linking different data sources. For example, the combination of wedding date and birth date may allow for the re-identification of a particular person. *Anonymization* of data refers to de-identification that is irreversible: re-identification is impossible. A range of de-identification methods is available: removing variables, randomization, hashing, shuffling, sub-sampling, aggregation, truncation, generalization, adding noise, etc. Adding some noise to a continuous variable or the coarsening of values may have a limited impact on the quality of analysis results while ensuring confidentiality.

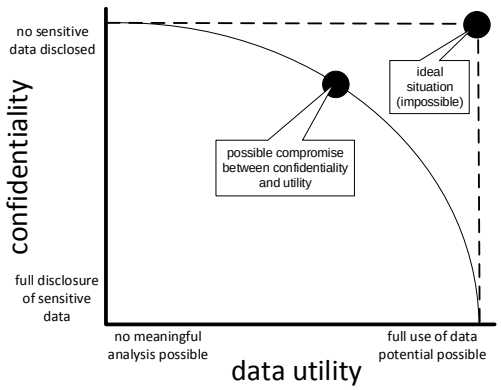


Figure 3: Tradeoff between confidentiality and utility.

There is a trade-off between minimizing the disclosure of sensitive information and the usefulness of analysis results (see Figure 3). Removing variables, aggregation, and adding noise can make it hard to produce any meaningful analysis results. Emphasis on confidentiality (like security) may also reduce convenience. Note that *personalization often conflicts with fairness and confidentiality*. Disclosing all data, supports analysis, but jeopardizes confidentiality.

Access rights to the different types of data and analysis results in the "data science pipeline" (Figure 1) vary per group. For example, very few people will have access to the "raw data" stored in the information system ❷. More people will have access to the data used for analysis and the actual analysis results. Poor cybersecurity may endanger confidentiality. Good policies ensuring proper authentication (Are you who you say you are?) and authorization (What are you allowed to do?) are needed to protect access to the pipeline in Figure 1. Cybersecurity measures should not complicate access, data preparation, and analysis; otherwise people may start using illegal copies and replicate data.

See (Monreale et al., 2014; Nelson, 2015; President's Council, 2014) for approaches to ensure confidentiality.

2.3 Accuracy - Data Science without Guesswork: How to Answer Questions with a Guaranteed Level of Accuracy?

Increasingly decisions are made using a combination of algorithms and data rather than human judgement. Hence, analysis results need to be *accurate* and should not deceive end-users and decision makers. Yet, there are several factors endangering accuracy.

First of all, there is the problem of overfitting the data leading to "bogus conclusions". There are numerous examples of so-called *spurious correlations* illustrating the problem. Some examples (taken from (Vigen, 2015)):

- The per capita cheese consumption strongly correlates with the number of people who died by becoming tangled in their bedsheets.
- The number of Japanese passenger cars sold in the US strongly correlates with the number of suicides by crashing of motor vehicle.
- US spending on science, space and technology strongly correlates with suicides by hanging, strangulation and suffocation.
- The total revenue generated by arcades strongly correlates with the number of computer science doctorates awarded in the US.

When using many variables relative to the number of instances, classification may result in complex rules overfitting the data. This is often referred to as the *curse of dimensionality*: As dimensionality increases, the number of combinations grows so fast that the available data become sparse. With a fixed number of instances, the predictive power reduces as the dimensionality increases. Using cross-validation most findings (e.g., classification rules) will get rejected. However, if there are many findings, some may survive cross-validation by sheer luck.

In statistics, Bonferroni's correction is a method (named after the Italian mathematician Carlo Emilio Bonferroni) to compensate for the problem of multiple comparisons. Normally, one rejects the null hypothesis if the likelihood of the observed data under the null hypothesis is low (Casella and Berger, 2002). If we test many hypotheses, we also increase the likelihood of a rare event. Hence, the likelihood of incorrectly rejecting a null hypothesis increases (Miller, 1981). If the desired significance level for the whole collection of null hypotheses is α, then the Bonferroni correction suggests that one should test each individual hypothesis at a significance level of $\frac{\alpha}{k}$ where k is the number of null hypotheses. For example, if $\alpha = 0.05$ and $k = 20$, then $\frac{\alpha}{k} = 0.0025$ is the required significance level for testing the individual hypotheses.

Next to overfitting the data and testing multiple hypotheses, there is the problem of *uncertainty in the input data* and the problem of *not showing uncertainty in the results*.

Uncertainty in the input data is related to the fourth "V" in the four "V's of Big Data" (Volume, Velocity, Variety, and Veracity). Veracity refers to the trustworthiness of the input data. Sensor data may be uncertain, multiple users may use the same account, tweets may be generated by software rather than people, etc. These uncertainties are often not taken into account during analysis assuming that things "even out" in larger data sets. This does not need to be the case and the reliability of analysis results is affected by unreliable or probabilistic input data.

When we say, "we are 95% confident that the true value of parameter x is in our confidence interval $[a,b]$", we mean that 95% of the hypothetically observed confidence intervals will hold the true value of parameter x. Averages, sums, standard deviations, etc. are often based on sample data. Therefore, it is important to provide a confidence interval. For example, given a mean of 35.4 the 95% confidence interval may be $[35.3, 35.6]$, but the 95% confidence interval may also be $[15.3, 55.6]$. In the latter case, we will interpret the mean of 35.4 as a "wild guess" rather than a representative value for true average value. Although we are used to confidence intervals for numerical values, decision makers have problems interpreting the expected accuracy of more complex analysis results like decision trees, association rules, process models, etc. Cross-validation techniques like k-fold checking and confusion matrices give some insights. However, models and decisions tend to be too "crisp" (hiding uncertainties). Explicit vagueness or more explicit confidence diagnostics may help to better interpret analysis results. Parts of models should be kept deliberately "vague" if analysis is not conclusive.

2.4 Transparency - Data Science that Provides Transparency: How to Clarify Answers Such that they become Indisputable?

Data science techniques are used to make a variety of decisions. Some of these decisions are made automatically based on rules learned from historic data. For example, a mortgage application may be rejected automatically based on a decision tree. Other decisions

According to *Bonferroni's principle* we need to avoid treating random observations as if they are real and significant (Rajaraman and Ullman, 2011). The following example, inspired by a similar example in (Rajaraman and Ullman, 2011), illustrates the risk of treating completely random events as patterns.
A *Dutch government agency is searching for terrorists by examining hotel visits* of all of its 18 million citizens (18×10^6). The hypothesis is that terrorists meet multiple times at some hotel to plan an attack. Hence, the agency looks for suspicious "events" $\{p_1, p_2\} \dagger \{d_1, d_2\}$ where persons p_1 and p_2 meet on days d_1 and d_2. How many of such suspicious events will the agency find if the behavior of people is completely random? To estimate this number we need to make some additional assumptions. On average, Dutch people go to a hotel every 100 days and a hotel can accommodate 100 people at the same time. We further assume that there are $\frac{18\times10^6}{100\times100} = 1800$ Dutch hotels where potential terrorists can meet.
The probability that two persons (p_1 and p_2) visit a hotel on a given day d is $\frac{1}{100} \times \frac{1}{100} = 10^{-4}$. The probability that p_1 and p_2 visit the *same* hotel on day d is $10^{-4} \times \frac{1}{1800} = 5.55 \times 10^{-8}$. The probability that p_1 and p_2 visit the same hotel on two different days d_1 and d_2 is $(5.55 \times 10^{-8})^2 = 3.086 \times 10^{-15}$. Note that different hotels may be used on both days. Hence, the probability of suspicious event $\{p_1, p_2\} \dagger \{d_1, d_2\}$ is 3.086×10^{-15}.
How many candidate events are there? Assume an observation period of 1000 days. Hence, there are $1000 \times (1000-1)/2 = 499{,}500$ combinations of days d_1 and d_2. Note that the order of days does not matter, but the days need to be different. There are $(18 \times 10^6) \times (18 \times 10^6 - 1)/2 = 1.62 \times 10^{14}$ combinations of persons p_1 and p_2. Again the ordering of p_1 and p_2 does not matter, but $p_1 \neq p_2$. Hence, there are $499{,}500 \times 1.62 \times 10^{14} = 8.09 \times 10^{19}$ candidate events $\{p_1, p_2\} \dagger \{d_1, d_2\}$.
The expected number of suspicious events is equal to the product of the number of candidate events $\{p_1, p_2\} \dagger \{d_1, d_2\}$ and the probability of such events (assuming independence): $8.09 \times 10^{19} \times 3.086 \times 10^{-15} = 249{,}749$. Hence, there will be around a quarter million observed suspicious events $\{p_1, p_2\} \dagger \{d_1, d_2\}$ in a 1000 day period!
Suppose that there are only a handful of terrorists and related meetings in hotels. *The Dutch government agency will need to investigate around a quarter million suspicious events involving hundreds of thousands innocent citizens.* Using Bonferroni's principle, we know beforehand that this is not wise: there will be too many false positives.

Example: Bonferroni's principle explained using an example taken from (Aalst, 2016). To apply the principle, compute the number of observations of some phenomena one is interested in under the assumption that things occur at random. If this number is significantly larger than the real number of instances one expects, then most of the findings will be false positives.

are based on analysis results (e.g., process models or frequent patterns). For example, when analysis reveals previously unknown bottlenecks, then this may have consequences for the organization of work and changes in staffing (or even layoffs). Automated decision rules (❻ in Figure 1) need to be as accurate as possible (e.g., to reduce costs and delays). Analysis results (❺ in Figure 1) also need to be accurate. However, accuracy is not sufficient to ensure acceptance and proper use of data science techniques. Both decisions ❻ and analysis results ❺ also need to be *transparent*.

Figure 4 illustrates the notion of transparency. Consider an application submitted by John evaluated using three data-driven decision systems. The first system is a black box: It is unclear why John's application is rejected. The second system reveals it's decision logic in the form of a decision tree. Applications from females and younger males are always accepted. Only applications from older males get rejected. The third system uses the same decision tree, but also explains the rejection ("because male and above 50").

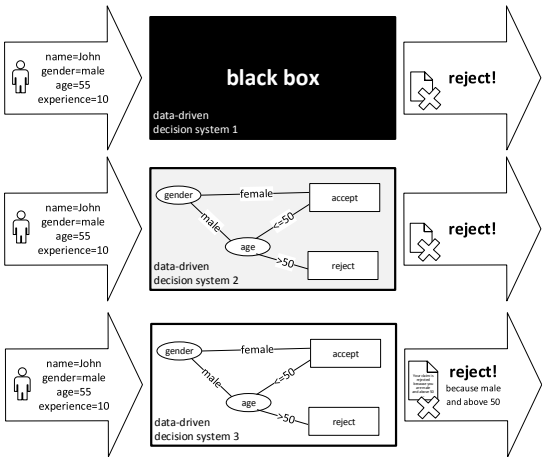


Figure 4: Different levels of transparency.

Clearly, the third system is most transparent. When governments make decisions for citizens it is often mandatory to explain the basis for such decisions.

Deep learning techniques (like many-layered neural networks) use multiple processing layers with

complex structures or multiple non-linear transformations. These techniques have been successfully applied to automatic speech recognition, image recognition, and various other complex decision tasks. Deep learning methods are often looked at as a "black box", with performance measured empirically and no formal guarantees or explanations. A many-layered neural network is not as transparent as for example a decision tree. Such a neural network may make good decisions, but it cannot explain a rule or criterion. Therefore, such black box approaches are non-transparent and may be unacceptable in some domains.

Transparency is not restricted to automated decision making and explaining individual decisions, it also involves the intelligibility, clearness, and comprehensibility of analysis results (e.g., a process model, decision tree, regression formula). For example, a model may reveal bottlenecks in a process, possible fraudulent behavior, deviations by a small group of individuals, etc. It needs to be clear for the user of such models (e.g., a manager) how these findings where obtained. The link to the data and the analysis technique used should be clear. For example, filtering the input data (e.g., removing outliers) or adjusting parameters of the algorithm may have a dramatic effect on the model returned.

Storytelling is sometimes referred to as "the last mile in data science". The key question is: How to communicate analysis results with end-users? *Storytelling is about communicating actionable insights to the right person, at the right time, in the right way.* One needs to know the gist of the story one wants to tell to successfully communicate analysis results (rather than presenting the whole model and all data). One can use natural language generation to transform selected analysis results into concise, easy-to-read, individualized reports.

To provide transparency there should be a clear link between data and analysis results/stories. One needs to be able to *drill-down* and inspect the data from the model's perspective. Given a bottleneck one needs to be able to drill down to the instances that are delayed due to the bottleneck. This related to *data provenance*: it should always be possible to reproduce analysis results from the original data.

The four challenges depicted in Figure 1 are clearly interrelated. There may be trade-offs between *fairness*, *confidentiality*, *accuracy* and *transparency*. For example, to ensure confidentiality we may add noise and de-identify data thus possibly compromising accuracy and transparency.

3 EXAMPLE: GREEN PROCESS MINING

The goal of *process mining* is to turn event data into insights and actions (Aalst, 2016). Process mining is an integral part of data science, fueled by the availability of data and the desire to improve processes. Process mining can be seen as a means to bridge the gap between data science and process science. Data science approaches tend to be process agonistic whereas process science approaches tend to be model-driven without considering the "evidence" hidden in the data. This section discusses challenges related to fairness, confidentiality, accuracy, and transparency in the context of process mining. *The goal is not to provide solutions, but to illustrate that the more general challenges discussed before trigger concrete research questions when considering processes and event data.*

3.1 What is Process Mining?

Figure 5 shows the "process mining pipeline" and can be viewed as a specialization of the Figure 1. Process mining focuses on the analysis of *event data* and analysis results are often related to *process models*. Process mining is a rapidly growing subdiscipline within both Business Process Management (BPM) (Aalst, 2013a) and data science (Aalst, 2014). Mainstream Business Intelligence (BI), data mining and machine learning tools are not tailored towards the analysis of event data and the improvement of processes. Fortunately, there are dedicated process mining tools able to transform event data into actionable process-related insights. For example, *ProM* (`www.processmining.org`) is an open-source process mining tool supporting process discovery, conformance checking, social network analysis, organizational mining, clustering, decision mining, prediction, and recommendation (see Figure 6). Moreover, in recent years, several vendors released commercial process mining tools. Examples include: *Celonis Process Mining* by Celonis GmbH (`www.celonis.de`), *Disco* by Fluxicon (`www.fluxicon.com`), *Interstage Business Process Manager Analytics* by Fujitsu Ltd (`www.fujitsu.com`), *Minit* by Gradient ECM (`www.minitlabs.com`), *myInvenio* by Cognitive Technology (`www.my-invenio.com`), *Perceptive Process Mining* by Lexmark (`www.lexmark.com`), *QPR ProcessAnalyzer* by QPR (`www.qpr.com`), *Rialto Process* by Exeura (`www.exeura.eu`), *SNP Business Process Analysis* by SNP Schneider-Neureither & Partner AG (`www.snp-bpa.com`), and *PPM webMethods Process Performance Manager* by Software AG (`www.softwareag.com`).

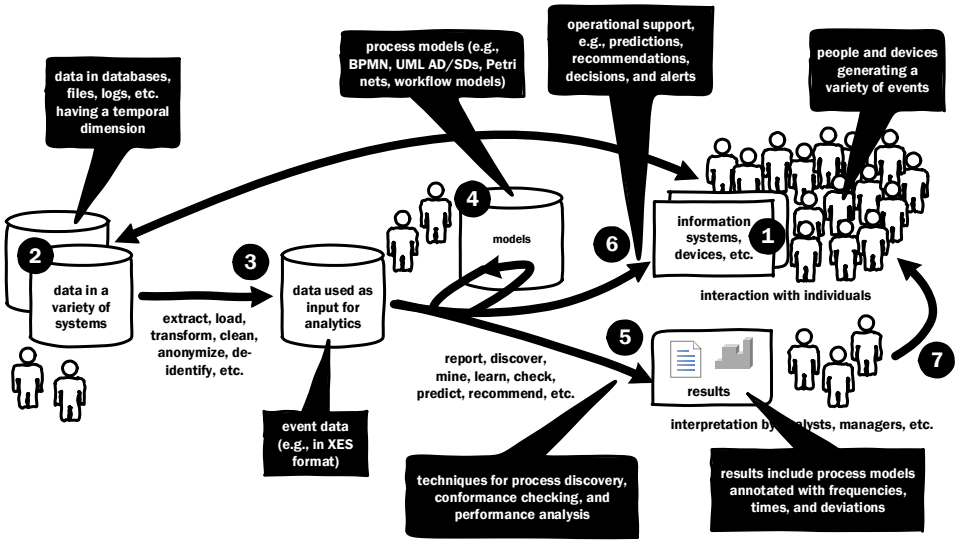


Figure 5: The "process mining pipeline" relates observed and modeled behavior.

3.1.1 Creating and Managing Event Data

Process mining is impossible without proper *event logs* (Aalst, 2011). An event log contains event data related to a particular process. Each event in an event log refers to one *process instance*, called *case*. Events related to a case are ordered. Events can have attributes. Examples of typical attribute names are activity, time, costs, and resource. Not all events need to have the same set of attributes. However, typically, events referring to the same activity have the same set of attributes. Figure 6(a) shows the conversion of an CSV file with four columns (case, activity, resource, and timestamp) into an event log.

Most process mining tools support XES (eXtensible Event Stream) (IEEE Task Force on Process Mining, 2013). In September 2010, the format was adopted by the IEEE Task Force on Process Mining and became the de facto exchange format for process mining. The IEEE Standards Organization is currently evaluating XES with the aim to turn XES into an official IEEE standard.

To create event logs we need to extract, load, transform, anonymize, and de-identify data in a variety of systems (see ❸ in Figure 5). Consider for example the hundreds of tables in a typical HIS (Hospital Information System) like ChipSoft, McKesson and EPIC or in an ERP (Enterprise Resource Planning) system like SAP, Oracle, and Microsoft Dynamics. Non-trivial mappings are needed to extract events and to relate events to cases. Event data needs to be scoped to focus on a particular process. Moreover, the data also needs to be scoped with respect to confidentiality issues.

3.1.2 Process Discovery

Process discovery is one of the most challenging process mining tasks (Aalst, 2011). Based on an event log, a process model is constructed thus capturing the behavior seen in the log. Dozens of process discovery algorithms are available. Figure 6(c) shows a process model discovered using ProM's *inductive visual miner* (Leemans et al., 2015). Techniques use Petri nets, WF-nets, C-nets, process trees, or transition systems as a representational bias (Aalst, 2016). These results can always be converted to the desired notation, for example BPMN (Business Process Model and Notation), YAWL, or UML activity diagrams.

3.1.3 Conformance Checking

Using conformance checking discrepancies between the log and the model can be detected and quantified by replaying the log (Aalst et al., 2012). For example, Figure 6(c) shows an activity that was skipped 16 times. Some of the discrepancies found may expose undesirable deviations, i.e., conformance checking signals the need for a better control of the process. Other discrepancies may reveal desirable deviations and can be used for better process support. Input for conformance checking is a process model having executable semantics and an event log.

3.1.4 Performance Analysis

By replaying event logs on process model, we can compute frequencies and waiting/service times. Using alignments (Aalst et al., 2012) we can relate cases to paths in the model. Since events have timestamps,

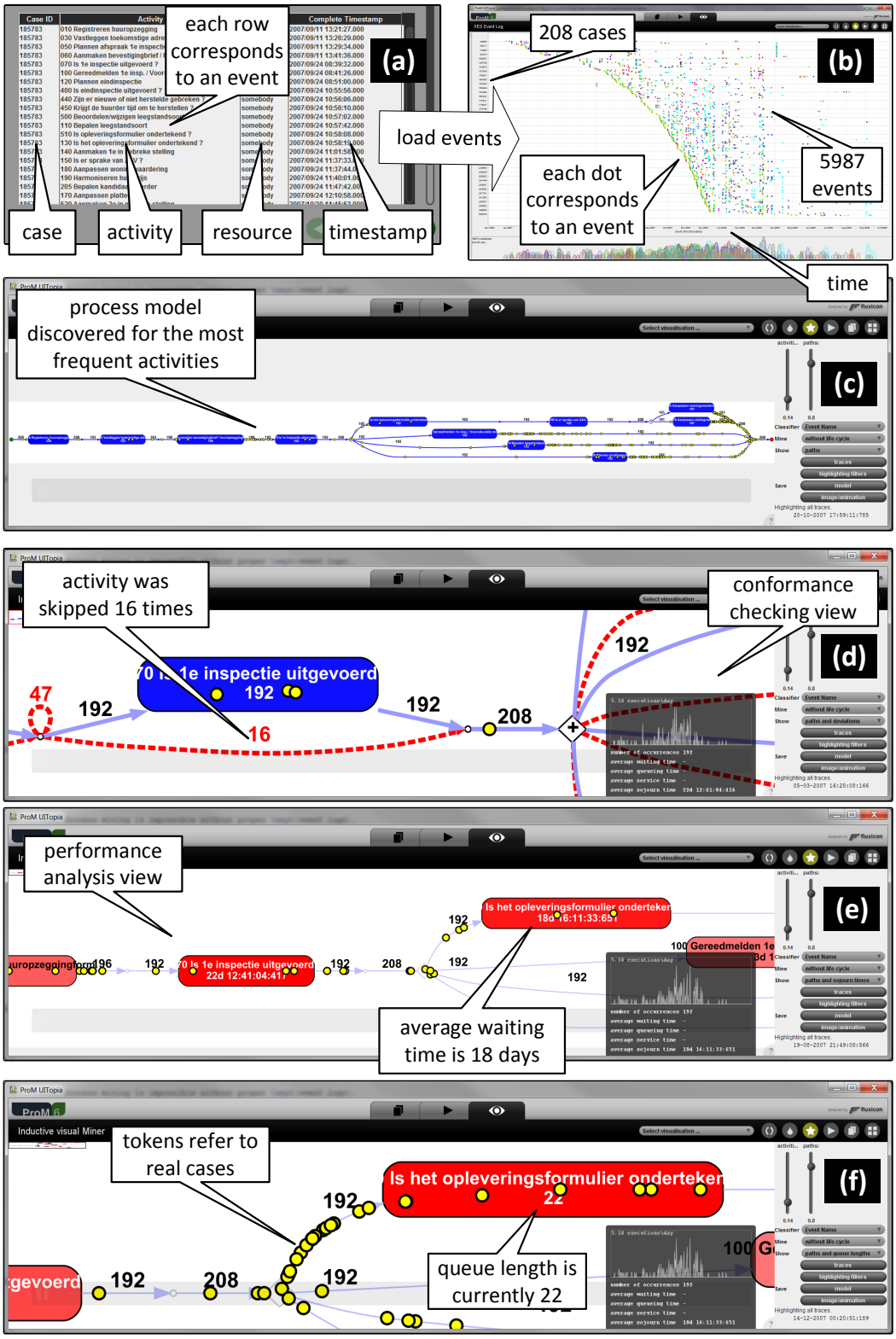


Figure 6: Six screenshots of ProM while analyzing an event log with 208 cases, 5987 events, and 74 different activities. First, a CSV file is converted into an event log **(a)**. Then, the event data can be explored using a dotted chart **(b)**. A process model is discovered for the 11 most frequent activities **(c)**. The event log can be replayed on the discovered model. This is used to show deviations **(d)**, average waiting times **(e)**, and queue lengths **(f)**.

we can associate the times in-between events along such a path to delays in the process model. If the event log records both start and complete events for activities, we can also monitor activity durations. Figure 6(d) shows an activity that has an average waiting time of 18 days and 16 hours. Note that such bottlenecks are discovered without any modeling.

3.1.5 Operational Support

Figure 6(e) shows the queue length at a particular point in time. This illustrates that process mining can be used in an online setting to provide operational support. Process mining techniques exist to predict the remaining flow time for a case or the outcome of a process. This requires the combination of a discovered process model, historic event data, and information about running cases. There are also techniques to recommend the next step in a process, to check conformance at run-time, and to provide alerts when certain Service Level Agreements (SLAs) are violated.

3.2 Challenges in Process Mining

Table 1 maps the four generic challenges identified in Section 2 onto the six key ingredients of process mining briefly introduced in Section 3.1. Note that both cases (i.e., process instances) and the resources used to execute activities may refer to individuals (customers, citizens, patients, workers, etc.). Event data are difficult to fully anonymize. In larger processes, most cases follow a unique path. In the event log used in Figure 6, 198 of the 208 cases follow a unique path (focusing only on the order of activities). Hence, knowing the order of a few selected activities may be used to de-anonymize or re-identify cases. The same holds for (precise) timestamps. For the event log in Figure 6, several cases can be uniquely identified based on the day the registration activity (first activity in process) was executed. If one knows the timestamps of these initial activities with the precision of an hour, then almost all cases can be uniquely identified. This shows that the ordering and timestamp data in event logs may reveal confidential information unintentionally. Therefore, it is interesting to investigate what can be done by adding noise (or other transformations) to event data such that the analysis results do not change too much. For example, we can shift all timestamps such that all cases start in "week 0". Most process discovery techniques will still return the same process model. Moreover, the average flow/waiting/service times are not affected by this.

Conformance checking (Aalst et al., 2012) can be viewed as a classification problem. What kind of cases deviate at a particular point? Bottleneck analysis can also be formulated as a classification problem. Which cases get delayed more than 5 days? We may find out that conformance or performance problems are caused by characteristics of the case itself or the people that worked on it. This allows us to discover patterns such as:

- Doctor Jones often performs an operation without making a scan and this results in more incidents later in the process.
- Insurance claims from older customers often get rejected because they are incomplete.
- Citizens that submit their tax declaration too late often get rejected by teams having a higher workload.

Techniques for *discrimination discovery* can be used to find distinctions that are not desirable/acceptable. Subsequently, techniques for *discrimination prevention* can be used to avoid such situations. It is important to note that discrimination is not just related to static variables, but also relates to the way cases are handled.

It is also interesting to use techniques from decomposed process mining or streaming process mining (see Chapter 12 in (Aalst, 2016)) to make process mining "greener".

For *streaming process mining* one cannot keep track of all events and all cases due to memory constraints and the need to provide answers in real-time (Burattin et al., 2014; Aalst, 2016; Zelst et al., 2015). Hence, event data need to be stored in aggregated form. Aging data structures, queues, time windows, sampling, hashing, etc. can be used to keep only the information necessary to instantly provide answers to selected questions. Such approaches can also be used to ensure confidentiality, often without a significant loss of accuracy.

For *decomposed/distributed process mining* event data need to be split based on a grouping activities in the process (Aalst, 2013b; Aalst, 2016). After splitting the event log, it is still possible to discover process models and to check conformance. Interestingly, the sublogs can be analyzed separately. This may be used to break potential harmful correlations. Rather than storing complete cases, one can also store shorter episodes of anonymized case fragments. Sometimes it may even be sufficient to store only *direct successions*, i.e., facts of the form "for some unknown case activity a was followed by activity b with a delay of 8 hours". Some discovery algorithms only use data on direct successions and do not require additional, possibly sensitive, information. Of course certain questions can no longer be answered in a reliable manner

Table 1: Relating the four challenges to process mining specific tasks.

	creating and managing event data	**process discovery**	**conformance checking**	**performance analysis**	**operational support**
fairness *Data Science without prejudice: How to avoid unfair conclusions even if they are true?*	The input data may be biased, incomplete or incorrect such that the analysis reconfirms prejudices. By resampling or relabeling the data, undesirable forms of discrimination can be avoided. Note that both cases and resources (used to execute activities) may refer to individuals having sensitive attributes such as race, gender, age, etc.	The discovered model may abstract from paths followed by certain under-represented groups of cases. Discrimination-aware process-discovery algorithms can be used to avoid this. For example, if cases are handled differently based on gender, we may want to ensure that both are equally represented in the model.	Conformance checking can be used to "blame" individuals, groups, or organizations for deviating from some normative model. Discrimination-aware conformance checking (e.g., alignments) needs to separate (1) likelihood, (2) severity and (3) blame. Deviations may need to be interpreted differently for different groups of cases and resources.	Straightforward performance measurements may be unfair for certain classes of cases and resources (e.g., not taking into account the context). Discrimination-aware performance analysis detects unfairness and supports process improvements taking into account trade-offs between internal fairness (worker's perspective) and external fairness (citizen/patient/customer's perspective).	Process-related predictions, recommendations and decisions may discriminate (un)intentionally. This problem can be tackled using techniques from discrimination-aware data mining.
confidentiality *Data Science that ensures confidentiality: How to answer questions without revealing secrets?*	Event data (e.g., XES files) may reveal sensitive information. Anonymization and de-identification can be used to avoid disclosure. Note that timestamps and paths may be unique and a source for re-identification (e.g., all paths are unique).	The discovered model may reveal sensitive information, especially with respect to infrequent paths or small event logs. Drilling-down from the model may need to be blocked when numbers get too small (cf. k-anonymity).	Conformance checking shows diagnostics for deviating cases and resources. Access-control is important and diagnostics need to be aggregated to avoid revealing compliance problems at the level of individuals.	Performance analysis shows bottlenecks and other problems. Linking these problems to cases and resources may disclose sensitive information.	Process-related predictions, recommendations and decisions may disclose sensitive information, e.g., based on a rejection other properties can be derived.
accuracy *Data Science without guesswork: How to answer questions with a guaranteed level of accuracy?*	Event data (e.g., XES files) may have all kinds of quality problems. Attributes may be incorrect, imprecise, or uncertain. For example, timestamps may be too coarse (just the date) or reflect the time of recording rather than the time of the event's occurrence.	Process discovery depends on many parameters and characteristics of the event log. Process models should better show the confidence level of the different parts. Moreover, additional information needs to be used better (domain knowledge, uncertainty in event data, etc.).	Often multiple explanations are possible to interpret non-conformance. Just providing one alignment based on a particular cost function may be misleading. How robust are the findings?	In case of fitness problems (process model and event log disagree), performance analysis is based on assumptions and needs to deal with missing values (making results less accurate).	Inaccurate process models may lead to flawed predictions, recommendations and decisions. Moreover, not communicating the (un)certainty of predictions, recommendations and decisions, may negatively impact processes.
transparency *Data Science that provides transparency: How to clarify answers such that they become indisputable?*	Provenance of event data is key. Ideally, process mining insights can be related to the event data they are based on. However, this may conflict with confidentiality concerns.	Discovered process models depend on the event data used as input and the parameter settings and choice of discovery algorithm. How to ensure that the process model is interpreted correctly? End-users need to understand the relation between data and model to trust analysis.	When modeled and observed behavior disagree there may be multiple explanations. How to ensure that conformance diagnostics are interpreted correctly?	When detecting performance problems, it should be clear how these were detected and what the possible causes are. Animating event logs on models helps to make problems more transparent.	Predictions, recommendations and decisions are based on process models. If possible, these models should be transparent. Moreover, explanations should be added to predictions, recommendations and decisions ("We predict that this case be late, because ...").

(e.g., flow times of cases).

The above examples illustrate that Table 1 identifies a range of novel research challenges in process mining. In today's society, event data are collected about anything, at any time, and at any place. Today's process mining tools are able to analyze such data and can handle event logs with billions of events. These amazing capabilities also imply a great responsibility. Fairness, confidentiality, accuracy and transparency should be key concerns for any process miner.

4 CONCLUSION

This paper introduced the notion of *"Green Data Science"* (GDS) from four angles: *fairness*, *confidentiality*, *accuracy*, and *transparency*. The possible "pollution" caused by data science should not be addressed (only) by legislation. We should aim for positive, technological solutions to protect individuals, organizations and society against the negative side-effects of data. As an example, we discussed "green challenges" in *process mining*. Table 1 can be viewed as a *research agenda* listing interesting open problems.

REFERENCES

Aalst, W. van der (2011). *Process Mining: Discovery, Conformance and Enhancement of Business Processes*. Springer-Verlag, Berlin.

Aalst, W. van der (2013a). Business Process Management: A Comprehensive Survey. *ISRN Software Engineering*, pages 1–37. doi:10.1155/2013/507984.

Aalst, W. van der (2013b). Decomposing Petri Nets for Process Mining: A Generic Approach. *Distributed and Parallel Databases*, 31(4):471–507.

Aalst, W. van der (2014). Data Scientist: The Engineer of the Future. In Mertins, K., Benaben, F., Poler, R., and Bourrieres, J., editors, *Proceedings of the I-ESA Conference*, volume 7 of *Enterprise Interoperability*, pages 13–28. Springer-Verlag, Berlin.

Aalst, W. van der (2016). *Process Mining: Data Science in Action*. Springer-Verlag, Berlin.

Aalst, W. van der, Adriansyah, A., and Dongen, B. van (2012). Replaying History on Process Models for Conformance Checking and Performance Analysis. *WIREs Data Mining and Knowledge Discovery*, 2(2):182–192.

Burattin, A., Sperduti, A., and Aalst, W. van der (2014). Control-Flow Discovery from Event Streams. In *IEEE Congress on Evolutionary Computation (CEC 2014)*, pages 2420–2427. IEEE Computer Society.

Calders, T. and Verwer, S. (2010). Three Naive Bayes Approaches for Discrimination-Aware Classification. *Data Mining and Knowledge Discovery*, 21(2):277–292.

Casella, G. and Berger, R. (2002). *Statistical Inference, 2nd Edition*. Duxbury Press.

European Commission (1995). Directive 95/46/EC of the European Parliament and of the Council on the Protection of Individuals with Wegard to the Processing of Personal Data and on the Free Movement of Such Data. Official Journal of the European Communities, No L 281/31.

European Commission (2015). Proposal for a Regulation of the European Parliament and of the Council on the Protection of Individuals with Wegard to the Processing of Personal Data and on the Free Movement of Such Data (General Data Protection Regulation). 9565/15, 2012/0011 (COD).

IEEE Task Force on Process Mining (2013). XES Standard Definition. www.xes-standard.org.

Kamiran, F., Calders, T., and Pechenizkiy, M. (2010). Discrimination-Aware Decision-Tree Learning. In *Proceedings of the IEEE International Conference on Data Mining (ICDM 2010)*, pages 869–874.

Leemans, S., Fahland, D., and Aalst, W. van der (2015). Exploring Processes and Deviations. In Fournier, F. and Mendling, J., editors, *Business Process Management Workshops, International Workshop on Business Process Intelligence (BPI 2014)*, volume 202 of *Lecture Notes in Business Information Processing*, pages 304–316. Springer-Verlag, Berlin.

Miller, R. (1981). *Simultaneous Statistical Inference*. Springer-Verlag, Berlin.

Monreale, A., Rinzivillo, S., Pratesi, F., Giannotti, F., and Pedreschi, D. (2014). Privacy-By-Design in Big Data Analytics and Social Mining. *EPJ Data Science*, 1(10):1–26.

Nelson, G. (2015). Practical Implications of Sharing Data: A Primer on Data Privacy, Anonymization, and De-Identification. Paper 1884-2015, ThotWave Technologies, Chapel Hill, NC.

Pedreshi, D., Ruggieri, S., and Turini, F. (2008). Discrimination-Aware Data Mining. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 560–568. ACM.

President's Council of Advisors on Science and Technology (2014). Big Data and Privacy: A Technological Perspective (Report to the President). Executive Office of the President, US-PCAST.

Rajaraman, A. and Ullman, J. (2011). *Mining of Massive Datasets*. Cambridge University Press.

Ruggieri, S., Pedreshi, D., and Turini, F. (2010). DCUBE: Discrimination Discovery in Databases. In *Proceedings of the ACM SIGMOD Intetrnational Conference on Management of Data*, pages 1127–1130. ACM.

Zelst, S. van, Dongen, B. van, and Aalst, W. van der (2015). Know What You Stream: Generating Event Streams from CPN Models in ProM 6. In *Proceedings of the BPM2015 Demo Session*, volume 1418 of *CEUR Workshop Proceedings*, pages 85–89. CEUR-WS.org.

Vigen, T. (2015). *Spurious Correlations*. Hachette Books.

The Power of Text Mining
How to Leverage Naturally Occurring Text Data for Effective Enterprise Information Systems Design and Use

Jan Vom Brocke
University of Liechtenstein, Liechtenstein

Abstract: This lecture shows how the design and use of Enterprise Information Systems can benefit from text mining techniques. It is estimated that more than 80 percent of today's data is stored in unstructured form (e.g., text, audio, image, video); and much of it is expressed in rich and ambiguous natural language. ERP systems store, for instance, piles of texts in the form of documents, E-mails, or service tickets. And comments on corporate social networks can be used to learn about work practices and preferences of users. We call this data naturally occurring data, since it is generated as a by-product of running IT-enabled business processes, and not provoked by analysts. Natural language processing has advanced over the past decades and today tools are available to analyze large amounts of unstructured texts in order to extract topics or sentiments buried in these. The talk gives examples on how successful companies leverage text mining to support important decisions as well as process, product and service innovations. It also shows how text mining can be applied as a new strategy of inquiry in EIS research, when e.g. online customer reviews are evaluated in order to learn about factors driving the ease of use or usefulness of EIS for specific user groups at specific times. The talk also presents a tool, which makes text mining accessible in the cloud both for practitioners and researcher, called MineMyText.com.

BRIEF BIOGRAPHY

Dr. Jan vom Brocke is professor for Information Systems at the University of Liechtenstein. He is the Hilti Chair of Business Process Management, Director of the Institute of Information Systems, and Vice President for Research and Innovation of the University of Liechtenstein. He has been elected Vice President of the Association for Information Systems (AIS), President of the Liechtenstein Chapter of the AIS and Vice President of the German Academic Association for Business Research (VHB). Jan has more than 15 years of experience in IS research, teaching and practice and he has published more than 300 papers in, among others, MIS Quarterly (MISQ), Journal of Management Information Systems (JMIS), Information & Management (I&M), and European Journal of Information Systems (EJIS). He is author and editor of 29 books, including the International Handbook on Business Process Management and the book BPM – Driving Innovation in a Digital World. He has taught at a number of universities, including the University of Muenster (Germany), the University of St.Gallen (Switzerland), and the LUISS University in Rome (Italy). Jan has held various editorial roles and leadership positions and he is an invited speaker and trusted advisor on IT and Management around the globe (see: www.janvombrocke.com).

Brocke, J.
The Power of Text Mining - How to Leverage Naturally Occurring Text Data for Effective Enterprise Information Systems Design and Use.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, page 23
ISBN: 978-989-758-187-8

DATABASES AND INFORMATION SYSTEMS INTEGRATION

FULL PAPERS

An Allegory on the Role of the Action Researcher to Enable User Engagement and Change Management in the Early Phases of Information Systems Implementation

Antonio Ghezzi
Department of Management, Economics and Industrial Engineering, Politecnico di Milano,
Via Lambruschini 4B, 20156 Milan, Italy
antonio1.ghezzi@polimi.it

Keywords: Information Systems, Alternative Genre, Enterprise Resource Systems, Allegory, User Engagement, Change Management, Action Research.

Abstract: Genres of communications significantly influence the evolution of a field of research. In the Information Systems (IS) domain, a debate has recently emerged on the chance to implement alternative genres to generate unconventional ways of looking at IS-related issues. This study hence proposes to apply allegory as an alternative genre to write publications accounting IS research. To exemplify the use of the allegory genre, the study tackles the role of the action researcher to enable user engagement and change management in the early phases of Information Systems implementation. The allegory is applied to the case of a Small-Medium-Enterprise undergoing ERP implementation. Reflecting on the allegory and its interpretation, it is argued that the action researcher can take a paramount role in IS change management as "user engagement enabler"; from a writing genre perspective, it is claimed that allegory is particularly suitable for writing action research accounts.

1 INTRODUCTION

The evolution of a field of research like that on Information Systems (IS) inherently relates not only to the content of investigation – in either its theoretical or empirical forms – and to the methodologies applied to conduct the research endeavor; it is also significantly shaped by the writing genre traditionally applied as a vehicle to report its content and findings.

In the last years, an intriguing debate has emerged with regards to the genres to be applied when writing academic publications (Rowe,2012). IS scholars and practitioners are currently discussing the opportunity to apply alternative genres in IS research representation. According to Mathiassen et al (2012), the term "alternative genres" refers to unconventional forms of thinking, doing, and communicating scholarship and practice. In particular, it is related to innovation with respect to epistemological perspectives, research methods, semantic framing, literary styles, and media of expression.

Provided that alternative genres are not valuable per se, but they become significant once they are fruitfully applied to writing studies on relevant IS issues, propose the adoption of alternative genres to tackle a significant problem in IS research: user engagement and change management in the early phases of Information Systems implementation – with specific reference to Enterprise Resource Planning (ERP) systems. To address this problem, I take the methodological perspective of an action researcher directly involved in the problem's observation and solution, and propose to employ the alternative genre of "allegory" to allegorically describe the role action researchers can play in enabling user engagement and change management in the early phases of ERP implementation.

Reflecting on the allegory and its interpretation, this study argues that the action researcher can take a paramount role in the IS change management process as "user engagement enabler"; from a writing genre perspective, the study also proposes that allegory can be beneficially applied as a genre to write action research accounts, due to the genre's peculiar characteristics.

Ghezzi, A.
An Allegory on the Role of the Action Researcher to Enable User Engagement and Change Management in the Early Phases of Information Systems Implementation.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 29-39
ISBN: 978-989-758-187-8

2 THEORY: CHANGE MANAGEMENT AND USER ENGAGEMENT IN IS IMPLEMENTATION

Change is an ever-present feature of organizational life both at an operational and strategic level (Burnes, 2004), and since information technology and organizational change show an inherent strong relationship (Markus and Robey, 1988), the issue of managing change determined by the introduction of new IS within an organizational setting has been a core theme in Information Systems (IS) research and practice (e.g. Aladwani, 2001; Lim et al., 2005)

In general terms, change management could be defined as "the process of continually renewing an organization's direction, structure, and capabilities to serve the ever-changing needs of external and internal customers" (Moran and Brightman, 2001). Both Organizational and IS theories widely recognize how Information Technology (IT) influences the nature of work, thus catalysing innovation while forcing incremental or radical organizational redesign (Thach and Woodman, 1994).

Through implementing IT, organizations aim at increasing process efficiency and effectiveness (with a possible beneficial impact on outward performance), although they also trigger inward organizational effects that mostly reflect on employees' routines, practices, habits and perceptions (Thach and Woodman, 1994): these non-trivial, subtle effects require dedicated effort to be understood and handled.

Focusing on the IS field, change management hence tackles the problem of how to govern the organizational transition determined by the introduction of new information technologies and systems (Markus and Robey, 1988).

Several studies have tackled the issue of user engagement in IS implementation, finding that such engagement is influenced by different factors. In his seminal work "Psychology of innovation resistance", Sheth (1981) argued that there are two main sources of resistance to IS innovations: perceived risk, which refers to one's perception of the risk associated with the decision to adopt the innovation; and habit, which refers to current practices that one is routinely doing. Joshi (1991) applied equity theory to IS implementation and found that individuals attempt to evaluate all changes on three levels: (i) gain or loss in their equity status; (ii) comparison between personal and organizational relative outcomes; and (iii) comparison between personal and other user's relative outcome in the reference group. They only resist to changes they see unfavourable, while changes that are favourable are sought after and welcomed. Gefen (2002) identified users' trust as a key determinant for their engagement in the complex process of ERP system customization: trust was increased when the vendor behaved in accordance with client expectations by being responsive, and decreased when it behaved in a manner that contradicted these expectations by not being dependable. Lim et al. (2005) investigated user adoption behavior and motivation dynamics of ERP systems from an expectancy perspective, and claimed that managerial actions shall target different levels of motivational factors to avoid counter-productive dissonances. Wang and Chen (2006) found that assistance of outside experts in ERP implementation is inevitable: competent consultants can facilitate communication and conflict resolution in the ERP consulting process and assist in improving ERP system quality.

Beyond identifying the factors behind user engagement, particularly relevant to this study are also two process models designed to obtain and enhance engagement.

According to organizational theory, change management aimed at cognitive redefinition of users' attitude and behaviour should follow a process model called the force field model, made of three stages (Schein and Bennis, 1965; Schein, 1999): (i) unfreeze the existing condition and apply a force to it in the attempt to motivate users to change; (ii) change and movement to a new state, by focusing on training and communication; and (iii) re-freeze to make new behaviours become habitual or institutionalized routines.

In assessing the complex social problem of users' resistance to ERP implementation, Aladwani (2001), elaborates on Sheth's (1981) model and proposed a process-oriented conceptual framework consisting of three phases: (i) knowledge formulation (where insight is gathered on needs, values, beliefs and interests of future IS users); (ii) strategy implementation (where change management leverages tools such as communication, endorsement and training to create awareness, stimulate feelings and drive adoption, by constantly confronting habits with perceived risks; and (iii) status evaluation (where the progress of ERP change management effort is monitored).

3 RESEARCH SETTING

The company this study considers is a Small Medium Enterprise (SME) operating as an artistic exhibition designer and manufactured, run and owned by a Chief Executive Officer who inherited it from his father. The company began operations in the early fifties and in 2012 it had gained worldwide recognition, being involved in several projects with renown institutions, such as the British Museum, the Tower of London Museum, the Louvre in Paris, the Museum of Modern Arts and the Metropolitan Museum in New York.

As the company grew globally, however, it was shaped by two diverging thrust: on the one hand, the CEO aimed at maintaining the company's inheritance of a SME and its craftsman approach towards each activity and work; on the other, a compelling need for organic development and structuration was perceived by the management. As a result, the organizational evolution was to some extent convoluted and not fully consistent: while some functions (e.g. design and manufacturing) operated with a high degree of structuration and technology support, others (e.g. administration, procurement, project management and marketing) were almost completely unstructured. Furthermore, Information Technology did not evolve alongside the company's manufacturing technologies. The little IT function was largely focusing on maintaining the computers used for running Computer Aided Design and Computer Aided Manufacturing software; data analysis and storing was either based on mere spreadsheets, or more frequently, on paperwork.

In late 2012, when the action research process began, it was time to make a strategic decision about IT. The management team had been consulting a shortlist of IT vendors for three months, and the most promising solution proposed was that of implementing an Enterprise Resource Planning (ERP) system to centralize and support information management and workflow throughout the functions. However, the CEO had profound doubts about this change, and his worries were somewhat justifiable. The CEO foresaw the introduction of such a pervasive system would determine radical modifications in several areas, with unpredictable results; he also he expected some of his employees to eventually resist to or impair the IT project. On top of this, he held a Philosophy and Literature background, which gave him an anti-conformist and original perspective on many strategic or organizational issues, including technology: he had contrasting feelings concerning IT, which he liked to philosophically define as "*a robot with huge potential to enhance human's capabilities, but after all, a robot with no will and no creative value in itself other than that of the human utilizing it"*.

The CEO's and his top management's primary concern was hence to adequately set and manage this IS transition.

4 ACTION RESEARCH METHODOLOGY

Action Research (AR) was primarily developed from the work of Kurt Lewin and his colleagues, and is based on a collaborative problem-solving relationship between the researcher and the client system, aiming at both managing change and generating new knowledge (Coghlan, 2000).

As a form of qualitative research (Myers, 1997), AR is described as a setting in which a client is involved in the process of data gathering, which is prevailingly under the charge of a researcher. Avison et al. (1999) define AR as an iterative process involving researchers and practitioners acting together on a particular cycle of activities, including problem diagnosis, action intervention, and reflective learning. According to Rapoport (1970), "action research aims to contribute both to the practical concerns of people in an immediate problematic situation and to the goals of social science by joint collaboration within a mutually acceptable ethical framework". Indeed, action Research is perhaps the most widely discussed collaborative research approach (see Baskerville and Wood-Harper 1998, Davison et al. 2004).

The collaboration this study depicts by means of the allegory alternative genre is set in an artistic exhibition design Small Medium Enterprise (SME) and began with the identification of a problem, i.e. the need to support the SME's CEO and Project Manager in enabling and managing change from a basic and piecemeal approach towards technology to the implementation of a broader ERP system. More specifically, the CEO and the Project Manager were concerned with user engagement, resistance to change and communication issues that could burden the early implementation phases.

This complex problem brought together multiple participants, all of whom had an interest in solving it. The set of participants included: Chief Executive Officer; the management team; the internal Project Manager; the SME's employees (also referred to as

users); the IT Vendor's Marketing Manager; and the team of three Action Researchers.

The problem that needed a solution was not easily solvable within the current community of practice inside of the company, who lacked specific IS and change management competencies, and furthermore called for the combination of knowledge from multiple perspectives, expertise, and disciplines (Mohrman et al., 2008). Hence, a problem-focused research approach like AR could provide a natural home for and evoke a need for collaboration that brought together multiple perspectives, including those of theory and practice. In part, this is because problems represent anomalies, and present a need to step outside of the daily reality that is driven by implicit theories, and to try to achieve a detachment that enables the search for new understandings that can guide action (Coghlan, 2000).

In order to solve the previously identified problem, from December 2011 to March 2012, the researchers who are authoring this study where directly involved in the early stages of the implementation process of an Enterprise Resource Planning system within the SME (thus following the direct involvement principle of the action research methodology), with the planned overarching objective to apply change management and organizational communication practices supporting the early phases of ERP implementation – with a focus on enhancing user engagement. Although the whole implementation project lasted till April, 2013, this study focuses on allegorically describing its first four months, where change management practices and user engagement dynamics where at the heart of the discussion.

The AR process was organized through a series of weekly meetings (for a total of 21 meetings, each lasting 2 hours 40 minutes on average) that the action researchers alternatively held with all the actors involved (including users). The content of such meetings was previously planned with and agreed upon by the CEO and the Project Manager, and these actors were open to the researchers' proposed lines of intervention. In the meetings, the action researcher set a flexible agenda, checked the progress status of previously identified actions, gathered insights from the participants, provided new content for discussion, set and explained new action points and assignments and instructed participants on how to act upon them.

In parallel, action researchers were involved in supporting the change management and communications activities and observing the user engagement process almost of a daily basis, in order to gather further information relevant to the research; they also operated "shoulder to shoulder" with the CEO and the Project Manager, and the result of this was that the researchers not only gained a deeper understanding of the company, its culture and its management's approach, but also gradually became accepted as a non-threatening and legitimate presence (Coghlan, 2000).

5 ALLEGORY AS ALTERNATIVE GENRE: DEFINITIONS, STRUCTURE AND PRINCIPLES

An allegory is "the representation of abstract ideas or principles by characters, figures, or events in narrative, dramatic, or pictorial form", and "a story, picture, or play employing such representation" (American Heritage Dictionary, 2011), where "the apparent meaning of the characters and events is used to symbolize a deeper moral or spiritual meaning" (Collins English Dictionary, 2003).

The term derives from the Greek allēgoría, derivative of allēgoreîn, i.e. to speak so as to imply something other. As a rhetorical device, an allegory is a figure of speech that makes wide use of metaphors (i.e. "a figure of speech in which a word or phrase is applied to an object or action that it does not literally denote in order to imply a resemblance" – Collins English Dictionary, 2003) and symbols (i.e. "something that represents or stands for something else, usually by convention or association, especially a material object used to represent something abstract" – Collins English Dictionary, 2003), though extending them to a complete and sense-making piece where complex ideas are illustrated by means of text or images that can be understood by the reader or viewer.

The very definition of allegory as a genre may be controversial. As the concept of genre represents a meaningful pattern of communication which consists of a sequence of speech acts (Yetim, 2006), and provided that "a genre is a category of art distinguished by a definite style, form or content" (American Heritage Dictionary, 2011), allegory is hard to fix since its convention are less formal or external, they are rather informal, skeletal or structural.

However, Quilligan (1979) in her book "The language of allegory: Defining the genre" argued that allegory is a genre, i.e. "a legitimate critical

category of a prescriptive status similar to that of the generic term 'epic'". Quilligan identifies the four main features that define the genre of allegory and its structure:

i.Text – the textual nature of the allegorical narrative, which unfolds as a series of punning commentaries related to one another;

ii. Pretext – which addresses the question of that source of which always stands outside any allegorical narrative and becomes the key to its interpretability (though not always to its interpretation). The relation between the text and the pretext is necessary slippery, yet by gauging its dimensions, we can begin to articulate the affinity of allegory as literary criticism to allegory as literary composition;

iii. Context – which addresses the question of formal evolution by tracking the cultural causes of allegory (allegories from different period may differ, since linguistic assumptions differed as well);

iv. Reader – which represents the final focus of any allegory, and the real action of any allegory is the reader's learning to read the text properly. "Other genres appeal to readers as human beings; allegory appeals to readers as readers of a system of signs, so it appeals to them in terms of their most distinguishing characteristics: as readers of, and therefore as creatures finally shaped by, their language" (Quilligan, 1979: 21).

The text and pretext hence focus on what the texts themselves say about the genre; the context provides the historical milieu out of which the author may write an allegory; and the reader is the ultimate producer of meaning (Nelson, 1968).

Considering that the primary characteristic of allegory as a genre is to separate the representation meaning from the inner and implied meaning, a mode of analysis for allegory can rely on hermeneutics (Myers, 1997). Hermeneutics is a classical discipline primarily concerned with the meaning of a text, and provides approaches to interpret it. The most common of such approaches is known as the "hermeneutic circle", which refers to the dialectic between the understanding of the text as a whole (a theory) and the interpretation of its details (single words), where the two dimensions are reciprocally validating and help deciphering the hidden meaning from the apparent meaning of narrative (Gadamer, 1976).

In the allegory this study presents, the whole story should be hermeneutically interpreted from the theoretical lenses of change management and user engagement in ERP implementation, while the details refer to specific aspects that influence and make sense within such context.

6 ALTERNATIVE GENRE APPLICATION: THE ALLEGORY OF THE SMALL VILLAGE

A small village was located in a wood and surrounded by a barriers of trees. The barrier was so thick nobody could actually see what was beyond it, and although it could be trespassed, no one had ever been bold enough to make the attempt. Rays of light made it through the ceiling of trees' branches, but branches were so many and intricate that the village was most of the time dark and surrounded by shades.

In the village lived a small community, who gathered to follow the lead of one whose visions were so fascinating and original that their heart was captured by them: he believed that human beings were meant to create works of art, and craftsmanship was mankind's deepest and essential virtue. The people from the village called him the Father, and once they stopped wandering in the dark of the wood to share his vision, the Father welcomed them in his community and taught them his idea of art as a form of beauty all men should pursue. From that time on, the Villagers' highest aspiration hence became to put such beautiful vision into practice.

They began collecting or even manufacturing tools they could use from what the wood offered them, and gathered into smaller groups of people whose abilities lied in one piece of art or another. As time went by, the Father selected a few chosen to help him lead his community that was growing, he called them the Wise Men and placed them at the lead of those smaller groups. The results of all their efforts were extraordinary, and notwithstanding the hardship they were confronted with, their masterful hands created objects of rare beauty.

Passers-by who were wandering nearby the village through the thick woods were fascinated by their works of art, and started asking for them: in return, they offered rewards coming from outside of the village they had been collected, and the village grew richer.

Word of the beauty of the crafts the community created spread, and soon many passers-by reached to the village to demand for the Villagers' pieces of art. At first, the Father and the Wise Men met these requests with joy, but soon they all realized the requests could not be met: the tools and instruments

their Villagers assembled to craft their art were incapable to perform the complex activities passers-by started asking for; and the wood, with his almost perennial darkness, was a difficult place to work in.

In the long nights in the wood, the Father tried to find a solution: however, his wisdom and art lied elsewhere, and the problem remained unanswered.

Then came the Wizard. He wore a cloak who concealed his figure, and he spoke a language no one in the village could understand. But he brought light: a light he could control, he could lit and stop at his will; a light Villagers could use to assemble new tools, to perform new works of art, and to illuminate the gloomy darkness of their village.

Still, the Wizard's mysterious light was met with doubt, or even fear: Villagers did not know where it came from, how to use it, and they were frightened by it. The Father perceived an inner power in that light, but it was something he could not fully comprehend himself: so he decided to host the Wizard in the village until he could unveil his mystery.

Some time passed, and a small group of Travelers, packed with big rucksacks on their shoulders, reached the village. These Travelers had seen some of the outer world and visited other villages before: but most shockingly, they seemed to understand part of what the Wizard was saying. While all other passers-by just came and went, the Father asked the Travelers to stay and help him disclosing the power of light.

The Travelers spent their days with the Father, to learn about the Villagers' habits; soon, they sympathized with them, and began understanding their fear for the new source of light, as well as their frustrations for the way they had been performing their activities till that day. The Travelers also attempted to speak with the Wizard, to understand his light's potential.

Villagers were afraid of relating with the Wizard, and were ashamed to talk to their Father about their dissatisfaction, but they felt they could confide in the Travelers and be open with them: after all, the Father introduced them, and it seemed a comfortable aura surrounded them.

Since the Father had many duties to perform as a leader of his community, he entitled a Wise Man to accompany the Travelers for all the time of their stay. The chosen Wise Man made sure all Villagers paid attention to the Travelers' questions and requests, and eventually learnt to understand some of the things the Wizard said or did.

It took many days to the Travelers to see, understand, reflect and learn; often, they were also seen walking around the village with awkward objects they pulled out of their rucksacks; but eventually they told the Father and his Wise Men that there was nothing to be feared about the light, although they needed them and all the Villagers to see this with their own eyes. And the Father agreed.

First, the Travelers convinced the Wizard to remove his cloak, to show everyone in the village he was a man like all the others; then, they helped him showing how the light could be used in the village to help or change what Villagers currently did. A big brazier was placed in the center of the village, and the light coming from it was strong and warm; the brazier could be a main source of light, but many other lights could be lit from it, and they could be used by the smaller groups of Villagers to perform their specific activities, shining from darkness; also, that light could alter forever the way the Villagers crafted their beautiful objects.

The Villagers were indeed impressed, but many of them were still frightened. The light could burn, they were used to darkness, and they had been using their skills in a certain fashion since they first joined the community. The Travelers hence knew that demonstrating the light's power was not enough: they needed to stay longer.

Almost each day since the brazier of light was brought in the village, the Travelers met with each Villagers, and then with the smaller groups of Villagers, reminding them of how dark their days were before the light came; they once again pulled some of their awkward objects out of their rucksacks and explained they came from their previous travels – many of them they even inherited form Travelers who lived in the past – and used them to show how the light helped others before the Villagers, and, by applying small changes to the objects, they could also show how the light could possibly help their own village. Then they asked the Villagers to tell stories on how the light could change their activities, the art they craft, and their lives, exposing their fears but also their hopes, and although several Villagers and even a few Wise Men were reluctant or shy to make up their own story, eventually the Father and the Travelers could convince them; and all these stories were reported to the Father and the Wizard, to make sure no voice would be left unheard. The Wizard was himself reluctant, as he could not see the reason why he should listen to the Villagers stories told in a different language than his own, but once again the Travelers were able to persuade him to change his perspective of reality and see it with the Villagers' eyes.

When a Villager complained or seemed to be left

apart, the Travelers spent time with her or him to understand the reasons, and all were treated the same way. The Travelers also got the Wizard to share his knowledge, and they translated while he taught the Villagers how to employ the light in many different ways. Those who proved remarkable skills at the new activities were also rewarded and indicated as examples to follow; some Villagers even passed from one smaller group to another. The Father and the Wise Men themselves showed passion and interest in these new activities, and took part to many of these gatherings.

Although the Villagers were still afraid of talking to the Wizard alone, they trusted the Travelers, since they never disguised themselves, they spoke a language similar to theirs, they listened to everyone and they had always treated everyone equally and fairly.

Once the lights were used everywhere in the village, the Father gathered all his community and said the dark age was over and would never return. A new era had started for those who lived in the village: craftsmanship had eventually found a new and more sophisticated instrument to be pursued.

The Travelers could hence leave the village, towards another endeavor.

7 DISCUSSION

7.1 Contribution to IS Practice

Applying the hermeneutical mode of text analysis (Gadamer, 1967) to the allegorical representation of ERP implementation allows to individuate two layers of meaning: (i) the apparent meaning, i.e., the way the narration is presented and appears as such; and (ii) the hidden meaning, i.e., the implied sense of the narration in the light of the IS issue tackled.

Table 1 shows the apparent and hidden meaning for each of the allegory's characters and elements.

Table 1: Apparent and hidden meanings in the small village allegory.

Apparent meaning	**Hidden meaning**
Allegory characters	
The Father	The CEO
The Wise Men	The Top Management
The chosen Wise Man	The Project Manager
The Villagers	The Employees
The Passers-by	The Customers
The Wizard	The ERP Vendor's Marketing Manager
The Travelers	The Action Researchers
Allegory elements	
Small village	SME
Wood	Environmental complexity
Barrier of trees	Closed approach
Darkness	Lack of technology
Works of art	SMEs products
Craftsmanship	Working skills
Smaller groups of villagers	SME's division of labor/functions
Rewards	Revenue streams
Wizard's language	IT language
Wizard's cloak	IT Professionals' different background
Villagers' language	Natural language
Light	Technology
Travelers' rucksacks	Action Researchers' theoretical background
Travelers' aura	Academic credibility
Travelers' objects pulled out of the rucksack	Action Researchers' theoretical models
Father's community duties	CEO's managerial tasks
Big brazier at the center of the village	ERP system
Smaller sources of light springing from the brazier	ERP modules supporting SME's functions
New instruments	New technological applications
Villagers', Wise Men's and Wizard's reluctance and shyness	Communication resistance to storytelling

The action research methodology and the change management theory provide the theoretical framing to decipher the hidden meaning of the allegory, whose implications for IS practice are various.

The allegory shows how action researchers acted in the empirical setting of a SME where the introduction of an ERP system was determining significant changes in the way users organized and performed their work and interpreted their organizational self.

The company was held together by the CEO's passion and eclectic leadership, although it started encountering significant issues as demand increased and became varied; moreover, the technological skills at hand were insufficient to govern a growingly complex company, but the CEO and his Top Management had little or no knowledge of IT. They perceived the opportunity represented by the ERP system, but were not capable of grasping it and a management-vendor leap appeared: this situation was similar to what Wang and Chen (2006) reported, where the lack of internal IT skills makes way for

external support. However, instead of looking for external consultancy firms or vendors to obtain such support, the company's CEO turned to action researchers. The involvement of action researchers in the project hence came with several advantages, and their role was crucial in key stages of the change management and user engagement process.

Action researchers first acted to demystify the new IS, by supporting the IT vendor in translating the IT language into natural language users could understand; by being almost ever-present they were responsive, and made sure the IT vendor removed his cultural "cloak" to become dependable and trustworthy (Gefen, 2002). Because of their academic status, an "aura" of credibility surrounded them from the management's and the users' standpoint, so they were seen as a much more reliable listeners than the IT vendor himself or any external consultancy firm could ever be: this aspect paved the way for open discussion, communication and sharing, all key elements in change management (Schein and Bennis, 1965; Gallivan and Keil, 2003).

Action researchers also played an intermediate role between the CEO, the Project Manager and users. They received endorsement from the CEO and worked shoulder to shoulder with him and the Project Manager to govern the change, so that the management could keep indirect control over the IS implementation's early stages without the risk to either abandon other managerial tasks supporting the business as usual (the "community duties") or be perceived as poorly committed to the innovation taking place; the researchers also had the CEO and the Top Management be involved in milestone steps of the project (e.g. kick-off meeting and regular meeting) and play as committed "ERP champions" to boost motivation for user adoption (Lim et al., 2005; Brown and Jones, 1998). Users did not enjoy complaining with their managers, and appreciated the role of the action researchers as trusted third parties they could rely on, as they perceived the researchers could collect their thoughts and feelings, relate them, add their own expertise and present them to the CEO and Project Manager in an organized, sound and apparently impartial mode.

Action researchers performed in a way that aimed at closing all the communication leaps and lapses (Gallivan and Keil, 2003) at three levels: (i) users-management; (ii) users-IT vendor; and (ii) user group-user group. In this process, action researchers became a sort of central buffer between the "Father" and the "Wise Men", the "Villagers" and the "Wizard", to solve all possible controversies arising. Consistently with the tenets of the equity-implementation model (Joshi, 1991), action researchers took the role of "organizational equalizers" and used communication devices to support the idea that no inequalities or loss of equities were perpetrated, so that the transition could be accepted and welcomed, rather than resisted.

User engagement was a priority in the change management process, and action researchers acted following a contingent approach that mixed rationalism (e.g. IS and change management theories and models) and experiments (e.g. hands-on training, exemplification, learning by doing and trial-and error approach) on the basis of their acquired knowledge of the specific research setting (Saarinen and Vepsäläinen, 1993) to enable it. They based their actions on the constant confrontation of users' habits and perceived risks (Sheth, 1981) to drive ERP adoption.

They were eventually the main actors to trigger and govern the unfreezing, change, re-freezing stages of the force field process model (Schein and Bennis, 1965; Schein, 1999), by: (i) sympathetically and empathically gathering knowledge on the needs, values, beliefs and interests of future IS users (Aladwani, 2001), feeding dissatisfaction over the "dark days" when IT was not available while clearly illustrating the benefits of the new solution; (ii) providing constant communication support to the IT Vendor as he tangibly started introducing the ERP system in the company, while listening to the voices of the internal customers and taking an active role on training; and (iii) setting the basis for a re-freezing of the newly acquired routines into institutionalized practices that the top management agreed upon.

Most originally, this study illustrates how the CEO and action researchers made use of "storytelling" as a communication device to create shared consensus on the IS transition: employees/users were requested to express their working expectations and feelings related to the new IS, and this made for better interiorizing of change and reduced long-term resistance. By doing so, the CEO and the action researchers performed an interesting paradigm shift in the classical approach to change management (Kettinger and Grover, 1995): they created and inflated an initial "communication resistance" aimed to lessen the impact of any future "user resistance". As the allegory discloses, the process of approaching ERP implementation through personal stories created early inter and intra-organizational tensions, which, however, in the short term eased participation, involvement and commitment to use the newly introduced system. Storytelling could hence become

part of IS change management practice, as a valuable communication device to support the early unfreezing and knowledge formulation phases where information on the users' habits and perceived risks should be gathered.

7.2 Contribution to IS Publication

This study also suggests to employ allegory as an alternative writing genres in IS publication.

An allegory can contain several layers of meanings, thus making the narration multidimensional and flexible and allowing to hide a deeper moral behind a literal interpretation of the text. The work of the IS action researcher/writer to add these layers to the traditional representation of her or his studies (as commonly reported in IS case studies) certainly requires and additional narrative effort: however, such work also forces the writer to dialectically move from the meaning to its symbol, from the symbol to a whole metaphor and then to the extended metaphor represented by the allegory itself. In this dialectic and iterative process, the researcher/writer has the chance to: deeply elaborate and reflect on the field data he collected; resort to a combination of expertise, intuition and creativity to develop an enlightening sensitivity towards the IS problem investigated (e.g. IS change management); describe such problem in a lively way where the explicit and the implicit perspectives coexist and both add to the account; and encourage the reader to empathically embark in the same interpretation process.

Thus, the allegory genre stimulates the construction of many apparently different though integrative narrations that can help the reader in the gradual activity of disentangling multifaceted and multidimensional IS problems and discover the action researcher's findings. A sense of empathic "discovery" will then permeate the allegory and accompany the reader during the interpretation process, and this will make for better interiorizing of the inner meanings – that is, the study's findings.

Paradoxically, an allegory could hence tell more of a writer's insight, understanding and perspective on a given IS phenomenon than a plain case description would: the allegory has the power to manage and convey the action researcher's intended meaning and personal insights which would have largely been "lost in translation" in traditional scientific writings. By properly framing the allegory in a methodological and contextual background (like this study attempts to do, by presenting the IS change management and user engagement theory and the action research methodology), the researcher/writer could offer an hermeneutical tool, a key to help the reader to translate metaphorical concept into real-world IS phenomena and elements. The theoretical and methodological frame would hence serve as the allegory's pre-text and con-text to stimulate a profound understanding of the literal text (Quilligan, 1979).

Due to its peculiarity, the alternative genre of allegory could show further characteristics. It could provide a narrative language that is appealing for a wider range of readers (other than researchers or IS specialists), possibly enlarging the target audience of IS studies towards different disciplines like Management; it could leverage symbolism and metaphors to nuance critical messages (e.g. IT vendor's scarce dependability) and convey positive or negative messages (e.g. "light" and "darkness" equated to the presence or absence of technology) that stay with the reader; and it could eventually place the reader into a position of self-denying self-consciousness (Quilligan, 1979), where he is more open to discovery and learning of the allegory's moral.

This study contends that allegory as an alternative genre could be most indicated to report action research endeavors, considering this research methodology's inner characteristics. Action researchers' activity is inherently multi-layered (as the allegory is): action researchers mix observation and action, detachment and involvement, description and normativity; they need to craft a narrative that draws from multiple perspectives and possibly unifies them into a single narrative; and their role is intimately hermeneutical, as they strive to help interpreting details in the light of the whole and validate the whole by means of details. The "Travelers" undertake journeys not only from company to company, but also cross-domain travels from theory to practice (and back to theory), from literal meaning (i.e. empirical events) to hidden revelations (theoretical and practical implications). Eventually, they can provide the sound theoretical and pragmatic key to read the allegory, always keeping in mind that an invisible thread shall relate the metaphor and the case they experienced (see Table 1).

Exploiting allegory as an alternative genre would constitute a normative breach that enables IS publications based on action research cases to overcome the limitations of canonical scientific writing (i.e. constraints on figures of speech, rhetorical devices and styles available; structural rigidity; limited accountability of internal responses

and motives, and limited perception of the intentional state vs. external response dualism; limited empathy and involvement evoked in the reader), thus providing a truly multifaceted account of the "organizational drama" (Avital and Vandenbosch, 2000) behind IS adoption.

8 CONCLUSIONS

This study's possible contribution is twofold.

Concerning IS practice, the allegory shows that a contingent approach that combines communication, endorsement, cognitive understanding and training can enable change management where change is caused by IS implementation. The study also proposes to include "user storytelling" as a valuable communication device to help the management and the researchers reveal employees' habits and perceived risks related to technological change, while buying them in in an emotional and empathic way that helps leapfrogging traditional resistance to change.

The first core claim from this study is that Action Researchers can play a paramount role in enabling and governing IS change management and users engagement. The mediation between theoretical detachment and professional involvement that characterizes action researchers, together with the "aura" springing from their academic background, make them a trusted and dependable party users can refer to in the often painful change process. Action researchers can support the key stages of the change management cycle by means of proper instruments like communication, managerial endorsement and training supervision, combined with their theoretical and practical IS endowment, to create a comfort zone for users where awareness is increased, empathy is stimulated, conflicts are resolved and adoption is driven.

The second core claim this study presents is that allegory is an alternative genre that could be beneficially employed to account for action research endeavors. Allegory as a genre shows similarities with the action researcher's multi-layered and multidimensional activity, and could force the researcher/writer into a reflection, abstraction and transposition cycle that could support his elaboration of his study's findings. The risk action researchers run is to be so involved in the project they observe and operate in that they eventually become incapable to get detached from it and grasp its deeper findings (that may be hiding below the surface of the operational activities performed). Writing the action research account in the form of an allegory demands to reinterpret a factual case in the light of symbols and metaphors that should connect to reality, while offering the reader a set of interpretation lenses borrowed from IS theory and practice. The positive result of this process is an enhanced ability to highlight the story's findings. And the hidden meaning of the allegory, once revealed and made apparent to the reader through an hermeneutical text analysis, could also allow deeper interiorizing of such findings and meanings.

REFERENCES

Aladwani, A. M. (2001). Change management strategies for successful ERP implementation. *Business Process management journal*, *7*(3), 266-275.

American Heritage Dictionary of the English Language, Fifth Edition (2011). Houghton Mifflin Harcourt Publishing Company.

Avison, D. E., Lau, F., Myers, M. D., and Nielsen, P. A. (1999). Action research.Communications of the ACM, 42(1), 94-97.

Baskerville, R., and Wood-Harper, A. T. (1998). Diversity in information systems action research methods. *European Journal of Information Systems*, 7(2), 90-107.

Boje, D. M. (2001). *Narrative methods for organizational & communication research*. Sage.

Boland, R., and Schultze, U. (1996). Narrating Accountability: Cognition and the Production of the Accountable Self", in R. Munro and R. Mouritsen (eds), *Accountability: Power, Ethos and the Technologies of Managing*. London: International Thomson Business Press, 1996.

Brown, A., and Jones, M. (1998). Doomed to Failure: Narratives of Inevitability and Conspiracy in a Failed IS Project. *Organization Science* (19:1), pp. 73-88.

Burnes, B. (2004) Managing Change: A Strategic Approach to Organisational Dynamics, 4th edn (Harlow: Prentice Hall).

Coghlan, D. and Brannick, T. (2005) *Doing Action Research in Your Own Organization*, SAGE Publications, London.

Coghlan, D. (2000). Interlevel dynamics in clinical inquiry. Journal of Organizational Change Management, 13(2), 190-200.

Coghlan, D. (2011) Action Research: Exploring Perspectives on a Philosophy of Practical Knowing. *The Academy of Management Annals* 5(1), 53-87.

Collins English Dictionary – Complete and Unabridged (2003). HarperCollins Publishers, 2003.

Cortimiglia, M., Ghezzi, A. and Frank, A. (2015) Business Model Innovation and strategy making nexus: evidences from a cross-industry mixed methods study. *R&D Management*, DOI: 10.1111/radm.12113.

Czarniawska, B. (2004). *Narratives in social science*

research. Sage.

Davison R., Martinsons, M. and Kock, N. (2004) Principles of Canonical Action Research. *Information Systems Journal* 14(1), 65-86.

Gadamer, H.G. (1976). Philosophical Hermeneutics. Berkeley: University of California Press.

Gallivan, M., and Keil, M. (2003), The user-developer communication process: a critical case study. *Information Systems Journal* 13 (1), 37–68.

Gefen, D. (2002). Nurturing clients' trust to encourage engagement success during the customization of ERP systems. *Omega*, *30*(4), 287-299.

Ghezzi, A., Cortimiglia, M. and Frank, A. (2015) Strategy and business model design in dynamic Telecommunications industries: a study on Italian Mobile Network Operators. Technological Forecasting and Social Change Vol. 90, Part A, 346-354.

Ghezzi A., Georgadis M., Reichl P., Di-Cairano Gilfedder C., Mangiaracina R. and Le-Sauze N. (2013) Generating Innovative Business Models for the Future Internet. *Info* 15(4), 43-68.

Ghezzi, A., Mangiaracina R. and Perego, A. (2012) Shaping the E-Commerce Logistics Strategy: a Decision Framework, *International Journal of Engineering Business Managemen*t, Wai Hung Ip (Ed.), ISBN: 1847-9790, InTech.

Ghezzi, A., Renga, F., and Balocco, R. (2009) A technology classification model for Mobile Content and Service Delivery Platforms. In *Enterprise Information Systems* (pp. 600-614). Springer Berlin Heidelberg.

Joshi, K. (1991). A model of users' perspective on change: the case of information systems technology implementation. *Mis Quarterly*, 229-242.

Kettinger, W. and Grover, V. (1995) Toward a Theory of Business Process Change Management. *Journal of Management Information Systems* 12(1), 9-30.

Lanzara, G. F. (1991). Shifting stories. Learning from a reflective experiment in a design process. In *The reflective turn: Case studies in and on educational practice* (pp. 285-320). Teachers College Press New York.

Lim, E. T., Pan, S. L., and Tan, C. W. (2005). Managing user acceptance towards enterprise resource planning (ERP) systems–understanding the dissonance between user expectations and managerial policies. *European Journal of Information Systems*, *14*(2), 135-149.

Markus, M. L., and Robey, D. (1988). Information technology and organizational change: causal structure in theory and research. *Management science*, *34*(5), 583-598.

Mathiassen, L., Chiasson, M. and Germonprez, M. (2012) Style Composition in Action Research Publication. *MIS Quarterly* 36(2), 347-363.

Mohrman S.A, Pasmore W.A., Shani A.B. (Rami), Stymne B., Adler N. (2008) Toward Building a Collaborative Research Community, in: Shani A.B. (Rami), Mohrman S.A., Pasmore W.A., Stymne B., Adler N. (Eds.) Handbook of Collaborative Management Research, Thousand Oaks (CA): Sage.

Moran, J. W. and Brightman, B. K. (2001) 'Leading organizational change', Career Development International, 6(2), pp. 111–118.

Myers, M. D. (1997). Qualitative Research in Information Systems. *MIS Quarterly* (21:2), June 1997, pp. 241-242.

Quilligan, M. (1979). *The language of allegory: Defining the genre*. Cornell University Press.

Rapoport, R. (1970) Three Dilemmas of Action Research. *Human Relation*s 23(6), 499-513.

Rowe, F. (2012) Toward a richer diversity of genres in information systems research: new categorization and guidelines. *European Journal of Information Systems* 21, 469-478.

Saarinen, T., and Vepsäläinen, A. (1993). Managing the risks of information systems implementation. *European Journal of Information Systems*, *2*(4), 283-295.

Schein, E. and Bennis, W, (1965), Personal and Organizational Change Through Group Methods: The Laboratory Approach, New York: Wiley.

Schein, E. H. (1999). The corporate culture survival guide: Sense and nonsense about culture change. San Francisco: Jossey-Bass Publishers.

Sheth, J. (1981), "Psychology of innovation resistance", *Research inMarketing*, Vol. 4, pp. 273-82.

Thach, L., and Woodman, R. W. (1994). Organizational change and information technology: Managing on the edge of cyberspace. *Organizational Dynamics*, *23*(1), 30-46.

Wang, E. T., and Chen, J. H. (2006). Effects of internal support and consultant quality on the consulting process and ERP system quality. *Decision Support Systems*, *42*(2), 1029-1041.

Yetim, F. (2006) Acting with genres: discursive-ethical concepts for reflecting on and legitimating genres. *European Journal of Information Systems*, 15(1), 54–69.

Zwickl, P., Reichl, P. and Ghezzi, A. (2011) On the quantification of value networks: a dependency model for interconnection scenarios. In *Economics of Converged, Internet-Based Networks* (pp. 63-74). Springer Berlin Heidelberg.

The Data-driven Factory
Leveraging Big Industrial Data for Agile, Learning and Human-centric Manufacturing

Christoph Gröger[1,2], Laura Kassner[1], Eva Hoos[1], Jan Königsberger[1], Cornelia Kiefer[1], Stefan Silcher[1,3] and Bernhard Mitschang[1]
[1]*Graduate School of Excellence advanced Manufacturing Engineering, University of Stuttgart, Nobelstr.12, 70569 Stuttgart, Germany*
[2]*Robert Bosch GmbH, Robert-Bosch-Platz 1, 70839 Gerlingen-Schillerhöhe, Germany*
[3]*eXXcellent solutions GmbH, Heßbrühlstraße 7, 70565 Stuttgart, Germany*
{firstname.lastname}@gsame.uni-stuttgart.de, {firstname.lastname}@bosch.com, {firstname.lastname}@exxcellent.de

Keywords: IT Architecture, Data Analytics, Big Data, Smart Manufacturing, Industrie 4.0.

Abstract: Global competition in the manufacturing industry is characterized by ever shorter product life cycles, increasing complexity and a turbulent environment. High product quality, continuously improved processes as well as changeable organizational structures constitute central success factors for manufacturing companies. With the rise of the internet of things and Industrie 4.0, the increasing use of cyber-physical systems as well as the digitalization of manufacturing operations lead to massive amounts of heterogeneous industrial data across the product life cycle. In order to leverage these big industrial data for competitive advantages, we present the concept of the data-driven factory. The data-driven factory enables agile, learning and human-centric manufacturing and makes use of a novel IT architecture, the Stuttgart IT Architecture for Manufacturing (SITAM), overcoming the insufficiencies of the traditional information pyramid of manufacturing. We introduce the SITAM architecture and discuss its conceptual components with respect to service-oriented integration, advanced analytics and mobile information provisioning in manufacturing. Moreover, for evaluation purposes, we present a prototypical implementation of the SITAM architecture as well as a real-world application scenario from the automotive industry to demonstrate the benefits of the data-driven factory.

1 INTRODUCTION

Global competition in the manufacturing industry is characterized by ever shorter product life cycles, increasing complexity and a turbulent environment. High product quality, continuously improved processes as well as changeable organizational structures constitute critical success factors for manufacturing companies (Westkämper, 2014).

With the rise of the internet of things, initiatives like Industrie 4.0 (MacDougall, 2014), respectively Smart Manufacturing (Davis *et al.*, 2012), significantly foster the use of cyber-physical systems (CPS) (Shi *et al.*, 2011) as well as the digitalization of manufacturing operations and promote the vision of decentralized self-control and self-optimization of products and processes (Brettel *et al.*, 2014). This leads to enormous amounts of heterogeneous industrial data across the entire product life cycle, representing *big industrial data* (Kemper *et al.*, 2013). These data are both structured and unstructured, ranging, e.g., from machine sensor data on the shop floor to data on product usage as well as from data on customer complaints in social networks to data on failure reports of service technicians. Exploiting these data, that is, extracting valuable business insights and knowledge from these data, is one of the central challenges in Industrie 4.0 (Gölzer *et al.*, 2015). For example, these data can be used for optimization of product design, manufacturing execution and quality management.

However, the prevailing manufacturing IT architecture in practice, the information pyramid of manufacturing (ISA, 2000), prevents comprehensive data exploitation due to the following limitations: (1) complex point-to-point integration of heterogeneous IT systems limits a flexible integration of new data sources; (2) strictly hierarchical aggregation of information prevents a holistic view for knowledge extraction; (3) isolated information provisioning for the manufacturing control level and the enterprise control level impedes employee integration on the factory shop floor.

Gröger, C., Kassner, L., Hoos, E., Königsberger, J., Kiefer, C., Silcher, S. and Mitschang, B.
The Data-driven Factory - Leveraging Big Industrial Data for Agile, Learning and Human-centric Manufacturing.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 40-52
ISBN: 978-989-758-187-8

To address these issues, we present the concept of the *data-driven factory* which is based on the results of several research projects we have undertaken at the Graduate School of Excellence advanced Manufacturing Engineering (GSaME) at the University of Stuttgart in cooperation with various industry partners. The data-driven factory leverages big industrial data for *agile, learning and human-centric manufacturing* and makes use of a novel IT architecture, the *Stuttgart IT Architecture for Manufacturing (SITAM)*, overcoming the insufficiencies of the traditional information pyramid of manufacturing. The data-driven factory combines *service-oriented integration*, *advanced analytics* as well as *mobile information provisioning* in a holistic approach in order to exploit big industrial data for competitive advantages.

The remainder of this paper is organized as follows: First, we analyze the limitations of the information pyramid of manufacturing with respect to big industrial data and further discuss related work in Section 2. Next, we introduce the concept of the data-driven factory in Section 3 and derive technical requirements. Section 4 focuses on the SITAM architecture and its components in order to address these requirements and provide a technical framework for the data-driven factory. For evaluation purposes, we present a prototypical implementation of the SITAM architecture and discuss a real-world application scenario in Section 5 demonstrating the benefits of the data-driven factory. Finally, we conclude in Section 6 and highlight future work.

2 BIG INDUSTRIAL DATA AND THE INFORMATION PYRAMID OF MANUFACTURING

In this section, first, we analyze the limitations of the traditional information pyramid of manufacturing with respect to big industrial data in Section 2.1. Next, we discuss related work, especially recent manufacturing IT architectures addressing these limitations in Section 2.2.

2.1 Limitations of the Information Pyramid of Manufacturing

The information pyramid of manufacturing, also called the hierarchy model of manufacturing, represents the prevailing manufacturing IT architecture in practice (Vogel-Heuser *et al.*, 2009). It is used to structure data processing and IT systems in manufacturing companies and it is standardized in ISA 95 (ISA, 2000). In a simplified version, the information pyramid is comprised of three hierarchical levels (see Figure 1): the *enterprise control level* refers to all business-related activities and IT systems, such as enterprise resource planning (ERP) systems, the *manufacturing control level* focuses on manufacturing operations management especially with manufacturing execution systems (MES) and the *manufacturing level* refers to the machines and automation systems on the factory shop floor.

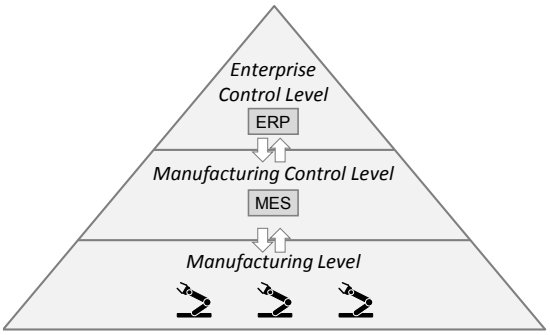


Figure 1: Information pyramid of manufacturing.

Data processing in the information pyramid is based on three fundamental principles (Vogel-Heuser *et al.*, 2009):

- *Central Automation* to control all activities top-down starting from the enterprise control level
- *Information Aggregation* to condense all data bottom-up starting from the manufacturing level
- *System Separation* to allow only IT systems at adjacent levels to directly communicate with each other

The digitalization of manufacturing operations as well as the massive use of CPS lead to big industrial data, i.e., enormous amounts of heterogeneous industrial data at all levels of the information pyramid and across the entire product life cycle (Kemper *et al.*, 2013). For instance, besides huge amounts of structured machine data and sensor data resulting from the shop floor, there are unstructured data on service reports and customer opinions in social networks. Exploiting these data, that is, extracting valuable business insights and knowledge, enables comprehensive optimization of products and processes (Gölzer *et al.*, 2015). For instance, customer satisfaction can be correlated with product design parameters using CAD data and CRM data or root causes of process quality issues can be analyzed using machine data and ERP data.

However, data processing according to the information pyramid of manufacturing prevents

comprehensive data exploitation due to the following major technical limitations (L_i):

- L_1: Central automation and system separation lead to a *complex and proprietary point-to-point integration of IT systems,* which significantly limits a flexible integration of new data sources across all hierarchy levels (Minguez *et al.*, 2010). For example, integrating an additional machine typically requires the costly and time-consuming adaptation of interfaces for a specific MES.
- L_2: Strictly hierarchical information aggregation leads to *separated data islands* preventing a holistic view for knowledge extraction (Kemper *et al.*, 2013). For instance, historic machine data at the manufacturing level is separated from ERP data at the enterprise control level, which prevents a holistic process performance analysis correlating, e.g., machine parameters and details on product configurations.
- L_3: Central control and information aggregation lead to *isolated information provisioning* focusing on the manufacturing control level and the enterprise control level and thus impede employee integration on the manufacturing level (Bracht *et al.*, 2011). For example, process execution data is typically aggregated for MES and ERP systems without information provisioning for shop floor workers.

To conclude, the function-oriented and strictly hierarchical levels of the information pyramid of manufacturing support a clear separation of concerns for the development and management of IT systems. However, the information pyramid lacks flexibility, holistic data integration and cross-hierarchical information provisioning. These factors significantly limit the exploitation of big industrial data and necessitate new manufacturing IT architectures, which are discussed in the following section.

2.2 Related Work: Manufacturing IT Architectures

We did a comprehensive literature analysis on recent architectural approaches for IT-based manufacturing. As result, we identified the following two major groups of work:

- *Abstract Frameworks for Industrie 4.0 and Smart Manufacturing,* which represent meta models and roadmaps for standardization issues, especially the Reference Architectural Model Industrie 4.0 (ZVEI, 2015) as well as the SMLC framework for Smart Manufacturing (Davis *et al.*, 2012)
- *Concrete Manufacturing IT Architectures,* which structure IT components and their relations in and across manufacturing companies on a conceptual level, especially (Vogel-Heuser *et al.*, 2009; Minguez *et al.*, 2010; Holtewert *et al.*, 2013; Papazoglou *et al.*, 2015)

The above frameworks are defined on a significantly higher abstraction level than the information pyramid of manufacturing. Hence, we concentrate on existing manufacturing IT architectures and analyze them with respect to the technical limitations of the information pyramid identified in Section 2.1. The common core of all of the above IT architectures is a service-oriented architecture (SOA) (Erl, 2008) in order to enable a flexible integration of IT systems – i.e. IT services – across all hierarchy levels (Minguez *et al.*, 2010; Holtewert *et al.*, 2013). In addition, in (Vogel-Heuser *et al.*, 2009), the need for a common data model standardizing the interfaces and the data of the IT services is underlined. In (Holtewert *et al.*, 2013; Papazoglou *et al.*, 2015), a marketplace with IT services is proposed in addition. In (Papazoglou *et al.*, 2015), a knowledge repository is part of the architecture. However, no concrete concepts for data integration, data analytics or data quality are presented.

All in all, these existing manufacturing IT architectures mainly address the limitation of a complex and proprietary point-to-point integration of IT systems in the information pyramid of manufacturing (L_1). Yet, they lack manufacturing-specific approaches for data analytics and information provisioning to fully address the limitations of separated data islands (L_2) as well as of isolated information provisioning (L_3). In contrast, our concept of the data-driven factory and the SITAM architecture address all three limitations in a holistic approach as detailed in the following sections.

3 THE DATA-DRIVEN FACTORY

The data-driven factory is a holistic concept to exploit big industrial data for competitive advantages of manufacturing companies. For this purpose, the data-driven factory addresses central economic challenges of today's manufacturing (Westkämper, 2014), particularly agility, learning ability as well as employee orientation, and makes use of a novel IT architecture, the Stuttgart IT Architecture for Manufacturing (SITAM), overcoming the insufficiencies of the traditional information pyramid of manufacturing.

The data-driven factory takes a holistic view on all data generated across the entire product life cycle,

from product design over manufacturing execution until service and support, including both structured data and unstructured data. Structured data generally refers to data in a relational form whereas unstructured data comprises text, audio and video files as well as images. In contrast to earlier integration approaches, especially Computer Integrated Manufacturing (Groover, 2008), the data-driven factory does *not* aim at totally automating all operations and decision processes but explicitly integrates employees in order to benefit from their knowledge, creativity and problem-solving skills.

In the following, we highlight the characteristics of the data-driven factory in Section 3.1 and derive corresponding technical requirements in Section 3.2 as a basis for the development of the SITAM architecture in Section 4.

3.1 Characteristics

From a manufacturing point of view, the data-driven factory is defined by the following core characteristics (see Figure 2):

- The data-driven factory enables *agile manufacturing* (Westkämper, 2014) by exploiting big industrial data for proactive optimization and agile adaption of activities. For instance, machine failures and turbulences can be predicted near real-time and manufacturing processes can be proactively adapted.
- The data driven factory enables *learning manufacturing* (Hjelmervik and Wang, 2006) by exploiting big industrial data for continuous knowledge extraction. For instance, concrete action recommendations can be learned from historic process execution data to optimize a specific metric, e.g., quality rate.
- The data driven factory enables *human-centric manufacturing* (Zuehlke, 2010) by exploiting big industrial data for context-aware information provisioning as well as knowledge integration of employees to keep the human in the loop. For example, shop floor workers are immediately informed about performance issues of the machine they are currently working at and can digitally create corresponding improvement suggestions, e.g., by recording a video.

To conclude, the data-driven factory leverages big industrial data for agile, learning and human-centric manufacturing. In this way, it creates new potentials for competitive advantages for manufacturing companies, especially with respect to efficient and at the same time agile processes, continuous and proactive improvement as well as the integration of knowledge and creativity of employees across the entire product life cycle.

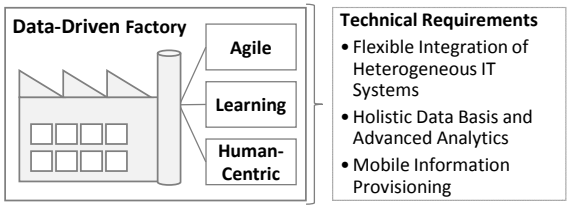


Figure 2: Characteristics and technical requirements of the data-driven factory.

3.2 Technical Requirements

Based on the above characteristics and taking into account the limitations of the information pyramid of manufacturing (see Section 2.1), we have derived the following technical core requirements (R_i) for the realization of the data-driven factory (see Figure 2):

- R_1: *Flexible Integration of Heterogeneous IT Systems* to rapidly include new data sources for agile manufacturing, e.g., when setting up a new machine
- R_2: *Holistic Data Basis and Advanced Analytics* for knowledge extraction in learning manufacturing, e.g., to prescriptively extract action recommendation from both structured and unstructured data
- R_3: *Mobile Information Provisioning* to ubiquitously integrate employees across all hierarchy levels for human-centric manufacturing, e.g., including service technicians in the field as well as product designers

In order to realize these requirements, a variety of IT concepts and technologies has to be systematically combined in an overall IT architecture. As we analyzed in Sections 2.1 and 2.2, the information pyramid of manufacturing lacks flexibility, holistic data integration and cross-hierarchical information provisioning (R_1-R_3). Thus, a novel manufacturing IT architecture is necessary, which is detailed in the next section.

4 SITAM: STUTTGART IT ARCHITECTURE FOR MANUFACTURING

The SITAM architecture is a conceptual IT architecture for manufacturing companies to realize the data-

driven factory. The architecture is based on the results and insights of several research projects we have undertaken in cooperation with various industry partners, particularly from the automotive and the machine construction industry.

In the following, we present an overview of the SITAM architecture in Section 4.1 and detail its components in Sections 4.2-4.6.

4.1 Overview

The SITAM architecture (see Figure 3) encompasses the entire product life cycle: *Processes*, *physical resources*, e.g., CPS and machines, *IT systems* as well as *web data sources* provide the foundation for several layers of abstracting and value-adding IT.The *integration middleware* (see Section 4.2) encapsulates these foundations into services and provides corresponding data exchange formats as well as mediation and orchestration functionalities.

The *analytics middleware* (see Section 4.3) and the *mobile middleware* (see Section 4.4) build upon the integration middleware to provide predictive and prescriptive analytics for structured and unstructured data around the product life cycle and mobile interfaces for information provisioning.

Together, the three middlewares enable the *composition of value-added services* for both human users and machines (see Section 4.5). In particular, services can be composed ad-hoc and offered as mobile or desktop apps on an *app marketplace* to integrate human users, e.g., by a mobile manufacturing dashboard with prescriptive analytics for workers. The added value from these services feeds back into the product life cycle for continuous proactive improvement and adaptation.

Cross-architectural topics (see Section 4.6) represent overarching issues relevant for all components and comprise *data quality*, *governance* as well as *security and privacy*.

In the following, the components of the SITAM architecture are described in greater detail.

4.2 Integration Middleware: Service-oriented Integration

The SITAM's *integration middleware* represents a changeable and adaptable integration approach, which is based on the SOA paradigm (Erl, 2008). The integration middleware is specifically tailored to manufacturing companies, providing the much needed flexibility and adaptability required in today's aforementioned turbulent environment with a permanent need of change.

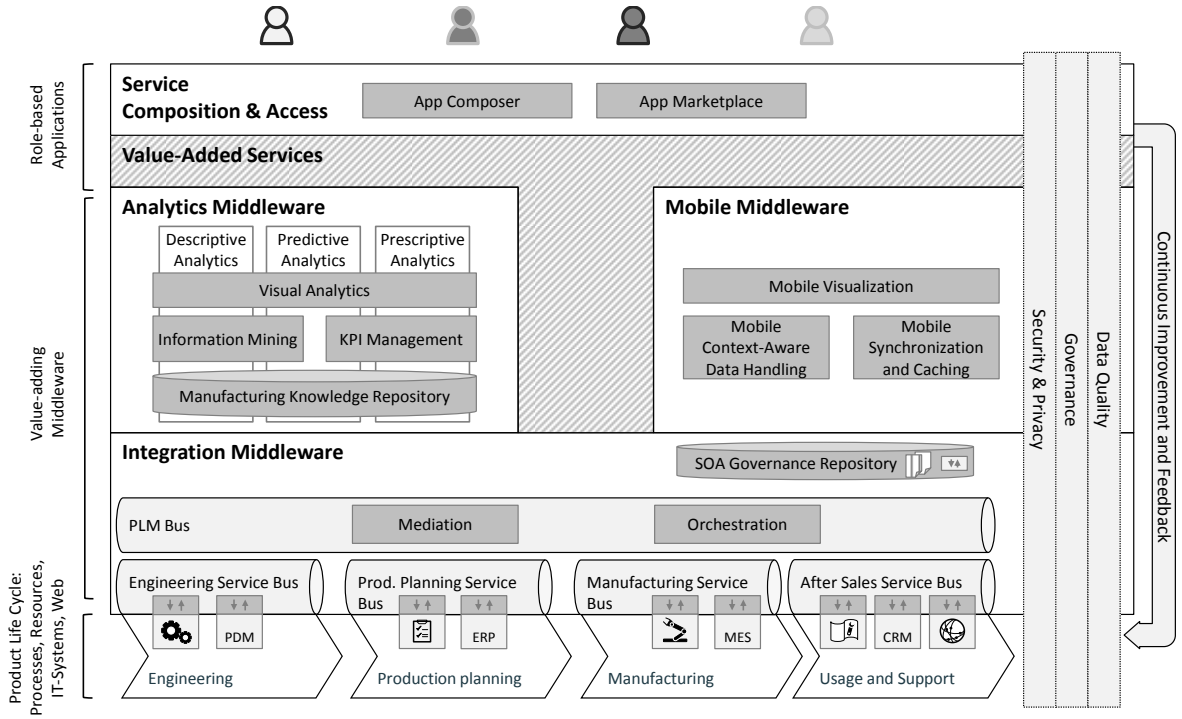


Figure 3: Overview of the Stuttgart IT Architecture for Manufacturing (SITAM).

To enable those benefits, it builds on a concept of hierarchically arranged *Enterprise Service Busses* (ESBs) following (Silcher *et al.*, 2013). Each one of these ESBs is responsible for the integration of all applications and services of a specific phase of the product life cycle.

All phase-specific ESBs are connected via a *superordinate Product-Lifecycle-Management-Bus* (PLM Bus). The PLM Bus is responsible for communication and mediation between phase-specific busses as well as for the orchestration of services.

This concept enables, for example, the easier integration of external suppliers without opening up too much of a company's internal IT systems to them by just "plugging" their own ESB into the PLM Bus. Besides, it also reduces the complexity by abstraction over the introduced integration hierarchy.

A dedicated sub-component providing real-time capabilities is used in the manufacturing phase to connect CPS and other real-time machine interfaces to the overall ESB compound.

The ESB hierarchy effectively abstracts and decouples technical systems and their services into a more business-oriented view, which we call *value-added services*. Value-added services use the basic services providing access to application data, orchestrate and combine them.

This decoupling also evens out different speeds in the development and change of applications or services. Companies often face the problem of having to integrate, e.g., legacy mainframe applications with modern mobile apps, which inherently have very different development speeds. By decoupling business-oriented services from the technical systems/services, each application can be developed separately and at its own pace, while the integration middleware handles all transformations and mediations that might be necessary to maintain compatibility.

Each phase-specific ESB also utilizes its own *phase-specific data exchange format* to handle the different requirements of each phase. For example, engineering has to be able to exchange large amounts of data, e.g., CAD models, whereas manufacturing requires the quick exchange of a large amount of smaller data chunks, e.g., MES production data. Aftersales on the other hand needs to handle both large CAD data as well as small, lightweight data structures, e.g., live car data.

The separation into different phase-specific ESBs allows each department or business unit to make use of specialized data exchange formats tailored to phase-specific needs.

To sum up, the hierarchical composition of phase-specific ESBs across the entire product life cycle and the changeable service-oriented abstraction of IT systems address requirement R_1 (flexible integration of heterogeneous IT systems) of the data-driven factory.

4.3 Analytics Middleware: Advanced Analytics

The analytics middleware is service-oriented and comprises several manufacturing-specific analytics components which are crucial for a data-driven factory: The *manufacturing knowledge repository* for storing source data and analytics-derived insights, *information mining* on structured and unstructured data, *management of key performance indicators* (KPIs), and *visual analytics*. The analytics middleware includes functionalities for descriptive, predictive and prescriptive analytics, with prescriptive analytics being a novel introduction which provides actionable problem solutions or preventative measures before critical conditions lead to losses (Evans and Lindner, 2012). In providing *integrative, holistic* and *near-real time analytics* on big industrial data of all data types, the SITAM analytics middleware transcends the analytics capabilities of existing approaches (see Section 2). This significantly contributes to the learning and agile characteristics of the data-driven factory.

Source data are extracted using predefined ETL functions from the integration middleware. Integrated data of structured and unstructured type from around the product life cycle are stored in the *manufacturing knowledge repository* along the lines of (Gröger *et al.*, 2014b) for maximum integration, minimum information loss and flexible access. Over the course of the product life cycle, this repository is enriched with various knowledge artefacts, e.g., analytics results like data mining models, business rules and free-form documents such as improvement suggestions. To store structured and unstructured source data in a scalable manner, the repository combines SQL and NoSQL storage concepts. It also includes the functionality for flexibly creating semantic links between source data and knowledge artefacts to support reasoning and knowledge management (see (Gröger *et al.*, 2014b)).

The *information mining* component can be subdivided into classical data mining and machine learning tools for structured data on the one hand, and tools for various types of unstructured data – text, audio, video – on the other hand.

We will discuss text analytics (Aggarwal and Zhai, 2012) in more detail since its use in a framework for integrative data analytics is novel and since text data harbor a wealth of hitherto untapped knowledge. Typically, text analytics applications

have been focused on one isolated unstructured data source and one analytical purpose, without integrating the results with analytics on structured data and with the disadvantage of information loss along the processing chain (Kassner *et al.*, 2014).

To secure flexibility of analytics and easy integration of data from different sources, we propose a set of basic and custom text analytics toolboxes, including domain-specific resources for the manufacturing and engineering domains and on an individual product domain level. This type of toolbox is similar to the generic and specific text analytics concepts proposed in (Kassner *et al.*, 2014). Value-added applications of these text analytics tools fall into two main categories: (1) information extraction tasks and (2) direct support of human labor through partial automation. For example, presenting the top ten errors for a specific time span based on text in shop floor documentation is an information extraction task which helps workers gain insights into weaknesses of the production setup. Using features of text reports, for example occurrences of particular domain-specific keywords, to predict the likelihood of certain error codes which a human expert must manually assign to these text reports, constitutes an example of a direct support analytics task (see (Kassner and Mitschang, 2016) for an implementation and proof of concept of this use case within the SITAM architecture).

Information mining can then be applied to discover knowledge, which is currently hidden in a combination of structured data and extracts from unstructured data. For example, process and machine data from the shop floor can be matched up with timestamps and extracted topics or relations from unstructured error reports to discover root causes for problems which have occurred. Real-time process data from the shop floor can be compared to historical data to discover indicators for problematic situations and prescribe measures for handling them, for example speeding up a machine when a delayed process has been discovered.

In order to constitute the backbone of a truly data-driven factory, information mining has to be conducted near real-time, on a variety of data sources as-needed, and manufacturing processes, sales, delivery, logistics and marketing campaigns have to adjust to meet the prescriptions derived from analytics results.

The *management of key performance indicators* is another important component and can be greatly improved by readily available and flexible analytics on a multitude of data sources. Instead of being an offline process conducted by the executive layer based on aggregated reporting data, KPI management can become a continuous and pervasive process, as data analytics feedback loops are in place for all processes around the product life cycle and at any level of the process hierarchy.

Finally, the analytics middleware also includes *visual analytics* for data exploration through human analysts: This type of analytics mainly combines information mining and visualization techniques to present large data sets to human observers in an intuitive way, allowing them to make sense of the data beyond the capabilities of analytics algorithms. Thus, visual analytics keep the human in the loop according to human-centric manufacturing.

Thus, the analytics capabilities of our reference architecture for the data-driven factory transcend those of related conceptual work in several aspects: (1) They include prescriptive, not just predictive or descriptive analytics, (2) they fully integrate structured and unstructured data beyond the manufacturing process, (3) they stretch across the entire product life cycle and provide a holistic view as well as holistic data storage, and (4) they are decentralized yet integrative, since analytics services are combined as-needed to answer questions or supervise processes and keep the human in the loop. Advanced analytics mostly contribute to the fulfillment of requirement R_2, but also R_3 and R_1 of the data-driven factory.

4.4 Mobile Middleware: Mobile Information Provisioning

The mobile middleware enables mobile information provisioning and mobile data acquisition by facilitating the development and integration of manufacturing-specific mobile apps. Mobile apps (Clevenger, 2011) are running on smart mobile devices, such as smartphones, tablets, and wearables, and integrate humans into the data-driven factory. Due to their high mobility, workers on the shop floor have to have access to the services of the factory *anywhere and anytime*, e.g., viewing near real-time information or creating failure reports on-the-go, supported by the mobile devices' cameras and sensors. Workers can also actively participate in the manufacturing process, e.g., they can control the order in which products are produced. Furthermore, mobile apps offer an intuitive *task-oriented touch-based* design and enable users to consume only relevant data. Mobile devices also allow for the collection of new kinds of data, e.g., position data or photos. This enables new kinds of services such as context-aware apps and augmented-reality apps (Hoos *et al.*, 2014).

However, the development of mobile apps differs from the development of stationary applications due to screen sizes, varying mobile platforms, unstable

network connections and other factors. In addition, manufacturing-specific challenges arise (Hoos *et al.*, 2014), e.g., due to the complex data structures as well as the high volume of data. In contrast to existing approaches (see Section 2.2), the mobile middleware addresses these manufacturing-specific needs.

The mobile middleware comprises three components: mobile context-aware data handling, mobile synchronization and caching as well as mobile visualization.

The *mobile context-aware data handling* component provides manufacturing-specific context models describing context elements and relations, e.g., on the shop floor, as well as efficient data transfer mechanism so that only relevant data in the current context is transmitted to the mobile device. For instance, a shop floor worker specifically needs information on the current machine he is working at.

The *mobile synchronization and caching* component supports offline usage of mobile apps. This is important because a network connection cannot always be guaranteed, particularly on the factory shop floor. The components offers mechanisms to determine which data should be cached using context information provided by the context models.

The *mobile visualization* component provides tailored visualization schemas for manufacturing data, e.g., for CAD product models. For example, it provides a visualization schema to represent a hierarchical product structure and to browse it via touch gestures. Various screen sizes and touch-based interaction styles are considered.

To sum up, the mobile middleware enables the integration of the human by supporting the development and integration of mobile apps. This is done by offering manufacturing-specific services for data handling and visualization. Thus, by addressing requirement R_3 (mobile information provisioning), the mobile middleware contributes to the human-centric characteristic of the data-driven factory, i.e., keeping the human in the loop.

4.5 Service Composition and Value-added Services

The service-based and integrative nature of the SITAM architecture allows it to provide value-added services in several ways. We define *value-added services* as services which provide novel uses and thus create value by transcending the limitations of the information pyramid of manufacturing (see Section 2.1): By providing flexible interfaces for data and service provisioning (addressing limitation L_1), by integrating, analyzing and presenting data from several phases around the product life cycle (addressing limitation L_2) and by providing access to information in all the contexts in which it is needed and in which the traditional model may fail to do so (addressing limitation L_3). The value-added services offered in the SITAM architecture cut across the architectural layers, packaging and combining functionalities of the integration middleware, the analytics middleware and the mobile middleware.

In the SITAM architecture, services are composed and adapted *on the basis of user roles* and the information needs and permissions associated with them. For example, a shop floor worker receives detailed alerts related to the process step he is responsible for, whereas his production supervisor is concerned with the aggregated state of the entire manufacturing process across all process steps.

Ad-hoc *service composition* is enabled by the *app composer*. The app composer offers this functionality for users in all roles, regardless of their educational background or their ability to code. For example, data sources and analytics services can be mashed up and composed via drag-and-drop in a graphic user interface. Atomic or composed services can then be offered and distributed as apps in the *app marketplace* for all types of devices, both stationary and mobile.

To sum up, flexible service composition contributes to the fulfillment of requirement R_1 (flexible integration of heterogeneous IT systems) and the provisioning of composed services as mobile apps helps to fulfill requirement R_3 (mobile information provisioning) of the data-driven factory.

4.6 Cross-architectural Topics

Security and privacy, governance and *data quality* are overarching topics which must be considered at all layers of the architecture: at the data sources, in analytics and mobile middleware as well as in the applications. In the following, we focus on *SOA governance* and *data quality* as they require specific concepts for the data-driven factory. For general security and privacy issues in data management, we refer the reader to (Whitman and Mattord, 2007).

The governance of complex service-oriented architectures is often neglected in existing manufacturing IT architectures, such as (Papazoglou *et al.*, 2015), even though a lack of governance is one of the main reasons for failing SOA initiatives (Meehan, 2014).

SOA governance covers a wide range of aspects (a list of key aspects can be found in (Königsberger *et al.*, 2014)). With more and more systems being integrated – especially CPS, but also for example social media services – it is becoming difficult to keep track

of planned changes to those systems and services. For this reason, service change management and service life cycle management governance processes track and report those changes to service consumers and providers, governed for example via consumer and stakeholder management processes.

When setting up those governance processes, it is important to keep them as lightweight and unobtrusive as possible in order to minimize complexity and managerial effort. To support this, the SITAM architecture contains a central *SOA governance repository,* which is built on a specific SOA governance meta model described in (Königsberger *et al.*, 2014). The repository uses semantic web technologies that allow for example the use of semantic reasoning to detect new dependencies or missing information. The SOA Governance Repository also contains service data as well as operations data, spanning and providing support during all phases of the service life cycle, and therefore also supporting novel software development concepts like DevOps.

Apart from SOA governance, the *need for high quality data* is a direct consequence of the concept of the data-driven factory. A data quality framework for the data-driven factory needs to enable data quality measurement and improvement (1) in near real-time (2) at all analysis steps from data source to user (3) for all types of data accumulating in the product life cycle, especially structured data as well as unstructured textual, video, audio and image data.

Existing data quality frameworks, e.g., (Wang and Strong, 1996; Sebastian-Coleman, 2013), fail to satisfy these requirements. Hence, we translate these requirements into an extended data quality framework, which allows a flexible composition of data quality dimensions (e.g., timeliness, accuracy, relevance and interpretability) at all levels of the SITAM architecture (see (Wang and Strong, 1996) for an example list of data quality dimensions). Furthermore, we define sets of concrete indicators considering data consumers at all levels, from data source to user, and we allow for near real-time calculation of data quality (e.g., the confidence or accuracy of machine learning algorithms, language of text and speech, author of data sources and the distribution of data points on a timeline). This makes the quality of data and of resulting analytics results transparent at all levels and therefore enables holistic data quality improvement.

To sum up, we have seen that SOA governance and data quality are crucial factors across all layers of the SITAM architecture. A flexible composition of IT systems and services can be offered using service-oriented architectures. But complex service-oriented architectures are prone to fail without systematic SOA governance. Besides, a holistic data quality framework forms the basis to measure and improve data quality from data source to user, including the generated analytics results.

5 IMPLEMENTATION AND EVALUATION

In the following, we present current work on the realization of the SITAM architecture in a prototypical implementation in Section 5.1. Moreover, we introduce a real-world application scenario from the automotive industry using the SITAM architecture in Section 5.2 and finally evaluate the benefits of the SITAM architecture and the concept of the data-driven factory in Section 5.3.

5.1 Implementation Strategy and Prototype

Our current prototype covers core components in every layer of the SITAM architecture, in particular with respect to analytics, governance, mobile and repository aspects. In the following, we sketch major solution details and technologies we utilized. The latter were chosen from the large available pool of free and open source software to underline the broad applicability of the SITAM architecture and make the implementation easily adaptable to various industrial real-world settings.

The *integration middleware* relies on WSO2's Application Server and Business Process Server, to realize the hierarchical ESB structure as well as the orchestration of basic services and mediation between phase-specific ESBs as described in (Silcher *et al.*, 2013). Services within the prototype are implemented as either conventional SOAP web services or REST services. Data exchange formats are realized as XSD documents and stored in the SOA governance repository. The repository itself relies, as mentioned in Section 4.6, on semantic web technologies, mainly the resource description framework (RDF) and provides a web-accessible as well as a Web Service interface as described in (Königsberger *et al.*, 2014).

In the *analytics middleware*, the manufacturing knowledge repository is implemented as a federation of a relational database and a NoSQL system – we used the content management system Alfresco CMS – to store structured and unstructured data. These systems are integrated by a specific link store using a graph database such as Neo4j. The information mining component includes tools from the

Apache UIMA framework (Ferrucci and Lally, 2004) for unstructured data analytics, with the uimaFit extension (Ogren and Bethard, 2009) for rapidly building analytics pipelines to allow for on-the-fly analytics service composition. Structured data mining capabilities are taken for instance from the WEKA data mining workbench (Hall *et al.*, 2009). On this basis, manufacturing-specific predictive and prescriptive analytics are realized using various data mining techniques, especially decision tree induction, as described in (Gröger *et al.*, 2014a, 2014b).

Regarding the *mobile middleware*, we implemented several mobile apps, e.g., a mobile analytics dashboard for shop floor workers (Gröger *et al.*, 2014b) and a mobile product structure visualizer for engineers. We have implemented native apps for Android and for Windows as well as platform independent web apps using standardized web technology such as HTML5.

An *app marketplace* and a graphical interface for intuitive access to the *app composer* are currently under development, with inspiration coming from mashup platforms (Daniel and Matera, 2014) and app generator tools, such as (Francese *et al.*, 2015).

5.2 Application Scenario: Quality Management and Process Optimization in the Automotive Industry

To demonstrate the concept of the data-driven factory as well as the SITAM architecture, we have cooperated with an OEM to develop a real-world application scenario for the automotive industry. The scenario focuses on quality management and process optimization as critical success factors for OEMs especially in the automotive premium segment.

An automotive manufacturer collects *big industrial data*, including structured sales and machine data, sensor and text data around the product life cycle. These data originally reside in isolated databases; for instance, text reports about product and part quality from development, production and aftersales are all gathered via different IT systems. To ensure a realistic representation of source data and processes, on the one hand, we take advantage of publicly available data sources, such as the records of automotive complaints covering the US market and maintained by the NHTSA (NHTSA, 2014). On the other hand, we make use of anonymized data and internal knowledge resources of our industry partner.

On this basis, the SITAM architecture is applied to exploit these data for quality management and process optimization. In the following, we give an overview of representative value-added services and role-based apps across the product life cycle which are enabled by the SITAM architecture (see Figure 4). We focus on car paint quality as a recurring example (all data samples in the following are fictitious for reasons of confidentiality).

During product development and testing, quality data are collected through the mobile *dev Q app* by engineers and test drivers on the go, including text reports and image material. The *aftersales Q app* is used to collect aftersales quality data for the warranty and recovery process of damaged car parts in the form of unstructured text reports (e.g., "customer states that car paint is coming off after washing", "flaking paint on fender during extreme summer heat"). It has different profiles for quality engineers (whose primary task is the definition of new error codes), for quality expert workers (whose task it is to assign error codes to damaged parts) and for executives (who are interested in comparing aggregated error code data over time). In addition, quality data come in the form of *customer complaints* and via *social media* crawling services.

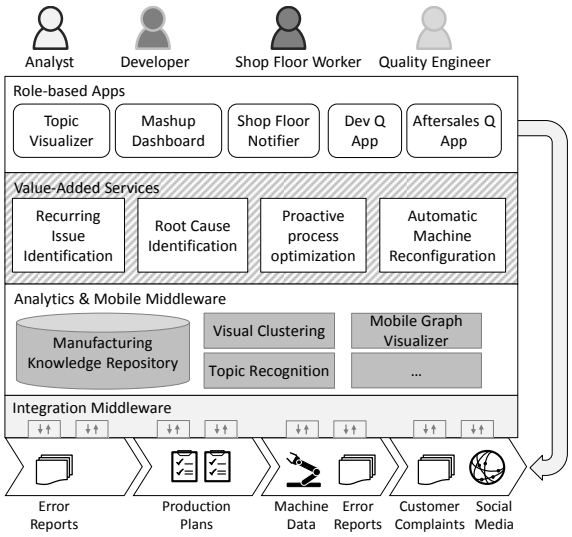


Figure 4: Value-added services and role-based apps in the application scenario.

After aggregating these data into the manufacturing knowledge repository via the integration middleware, *topic recognition* on the text data is performed as an information mining step. The topics (e.g., "paint flaking – heat", "paint damage – washing") are presented to a human *analyst* via *visual clustering* to pick the most pressing ones or perform minor reclassification. This constitutes a value-added service of *recurring issue identification* and is performed via the *topic visualizer app,* which makes use of the *mobile graph*

visualizer from the mobile middleware.

Next, the problem topics are combined with historical data from the production phase, especially machine data, shop floor environment data, and structured error counts for *root cause identification* (e.g., elevated humidity in the paint shop leading to a lower quality of paint and a higher risk of flaking when exposed to harsh environmental conditions). This analytics step is executed in an analytics and data *mashup dashboard* app, where data sources and analytics algorithms are combined ad-hoc, but can also be stored for recurring use.

Identified root causes and condition patterns serve as input for *proactive process optimization*. It makes use of prescriptive analytics to automatically identify potentially problematic situations (e.g., critical humidity in paint shops) during process execution and recommend actions to on-duty workers through a *shop floor notifier* app (e.g., to air the paint shops to decrease humidity) or trigger *automatic machine reconfiguration* (e.g., increasing air conditioning and heating to decrease humidity).

5.3 Evaluation and Benefits

Taking the above application scenario, we conceptually evaluate the SITAM architecture by analyzing the fulfillment of the technical requirements of the data-driven factory and contrasting it with the traditional information pyramid of manufacturing. Moreover, we summarize the resulting benefits of the data-driven factory.

In the application scenario, diverse systems across the product life cycle, such as machines, social media sources as well as sensors, are encapsulated as services and are uniformly represented in the SOA governance repository to ease integration and access in the integration middleware. By this service-oriented abstraction, the SITAM architecture enables a flexible integration of heterogeneous data sources as well as a flexible service composition fulfilling requirement R_1. This enables *agile manufacturing*, the first characteristic of the data-driven factory. Accessible service-based and role-based information provisioning also works towards keeping the human in the loop (*human-centric manufacturing)*. In contrast, a proprietary point-to-point integration according to the information pyramid of manufacturing would not scale up to the entire product life cycle in terms of complexity and costs.

To merge structured and unstructured data from different life cycle phases, e.g., aftersales quality data and machine data in the application scenario, all data are integrated in the manufacturing knowledge repository of the analytics middleware. Moreover, predictive and prescriptive analytics are provided, for instance, to derive action recommendations for process optimization according to the application scenario. Thus, the SITAM architecture provides a holistic data basis encompassing the product life cycle as well as advanced analytics for knowledge extraction fulfilling requirement R_2. This analytics capability provides functionalities for *learning manufacturing*, such as learned improvements for the quality-optimal design of both processes and products. It also is a prerequisite for agile process adaptations (*agile manufacturing*), such as the near real-time adaptation of production conditions to prevent known product quality issues. In contrast, the information pyramid of manufacturing is limited by separated data islands due to strictly hierarchical information aggregation.

In the application scenario, various mobile apps support seamless integration of employees, e.g., for data acquisition by test drivers using the dev Q app or for notifications of shop floor workers using the shop floor notifier. The mobile middleware facilitates the development of such manufacturing-specific apps using predefined manufacturing context models as well as specific visualization components, especially for product models. These apps can be easily deployed on various devices using the app marketplace. In this way, the SITAM architecture enables mobile information provisioning and fulfills requirement R_3 of the data-driven factory to ubiquitously integrate employees across all hierarchy levels. Thus, it provides the framework for *human-centric manufacturing* in keeping the human expert in the loop through data provisioning and data gathering. In contrast, central control and information aggregation lead to isolated information provisioning in the information pyramid of manufacturing.

To sum up, the SITAM architecture enables flexible system and data integration, advanced analytics and mobile information provisioning and thus fulfills all technical requirements (R_1-R_3) of the data-driven factory. In doing so, it exhibits the three characteristics of the data-driven factory, agile manufacturing, learning manufacturing and human-centric manufacturing.

6 CONCLUSION AND FUTURE WORK

In this article, we have presented the data-driven factory, an important contribution on the way to the realization of Industrie 4.0 and Smart Manufacturing. This concept completely alters the ways in which IT

systems are used and data are processed in manufacturing companies, thereby enabling *agile*, *learning* and *human-centric manufacturing* by leveraging *big industrial data*. The data-driven factory provides a stark contrast to the traditional information pyramid of manufacturing, which is fraught with the central weaknesses of proprietary point-to-point integration of IT systems, separated data islands and isolated information provisioning. Instead, the data-driven factory collects, analyzes and uses data holistically around the product life cycle and across all hierarchy levels of manufacturing. Thus, continuous data-driven optimization of processes and resources with the active participation of the 'human in the loop' is facilitated.

To realize the data-driven factory, we have developed the SITAM architecture which (1) flexibly integrates heterogeneous IT systems, (2) provides holistic data storage and advanced analytics covering the entire product life cycle, and (3) enables mobile information provisioning to empower human workers as active participants in manufacturing. We have prototypically implemented core components of the SITAM architecture in the context of a real-world application scenario concerned with quality and process management in the automotive industry. Our conceptual evaluation shows that the SITAM architecture enables the realization of the data-driven factory and the exploitation of big industrial data across the entire product life cycle.

In the future, we will extend our current prototype and further investigate the benefits of the data-driven factory on the example of additional industrial case studies, e.g., to concretize resulting competitive advantages in specific industries.

ACKNOWLEDGEMENTS

The authors would like to thank the German Research Foundation (DFG) as well as Daimler AG for financial support of this project as part of the Graduate School of Excellence advanced Manufacturing Engineering (GSaME) at the University of Stuttgart.

REFERENCES

Aggarwal, C.C. and Zhai, C.X. (2012), "An Introduction to Text Mining", in Aggarwal, C.C. and Zhai, C.X. (Eds.), *Mining Text Data*, Springer, Boston, pp. 1–10.

Bracht, U., Hackenberg, W. and Bierwirth, T. (2011), "A monitoring approach for the operative CKD logistics", *Werkstattstechnik*, Vol. 101 No. 3, pp. 122–127.

Brettel, M., Friederichsen, N., Keller, M. and Rosenberg, M. (2014), "How Virtualization, Decentralization and Network Building Change the Manufacturing Landscape: An Industry 4.0 Perspective", *International Journal of Science, Engineering and Technology*, Vol. 8 No. 1, pp. 37–44.

Clevenger, N.C. (2011), *iPad in the enterprise. Developing and deploying business applications*, Wiley, Indianapolis.

Daniel, F. and Matera, M. (2014), *Mashups - Concepts, Models and Architectures. Data-Centric Systems and Applications*, Springer, Heidelberg.

Davis, J., Edgar, T., Porter, J., Bernaden, J. and Sarli, M.S. (2012), "Smart Manufacturing, Manufacturing Intelligence and Demand- Dynamic Performance", *Computers & Chemical Engineering*, Vol. 47, pp. 145–156.

Erl, T. (2008), *SOA. Principles of service design, The Prentice Hall service-oriented computing series from Thomas Erl*, Prentice Hall, Upper Saddle River.

Evans, J.R. and Lindner, C.H. (2012), "Business Analytics: The Next Frontier for Decision Sciences", *Decision Line*, Vol. 43 No. 2, pp. 4–6.

Ferrucci, D. and Lally, D. (2004), "UIMA. An architectural approach to unstructured information processing in the corporate research environment", *Natural Language Engineering*, Vol. 10 No. 3-4, pp. 327–348.

Francese, R., Risi, M., Tortora, G. and Tucci, M. (2015), "Visual Mobile Computing for Mobile End-Users", *IEEE Transactions on Mobile Computing*, to appear.

Gölzer, P., Cato, P. and Amberg, M. (2015), "Data Processing Requirements of Industry 4.0 - Use Cases for Big Data Applications", in *Proceedings of the European Conference on Information Systems (ECIS) 2015,* Paper 61.

Gröger, C., Schwarz, H. and Mitschang, B. (2014a), "Prescriptive Analytics for Recommendation-Based Business Process Optimization", in *Proceedings of the International Conference on Business Information Systems (BIS) 2014*, Springer, Cham, pp. 25–37.

Gröger, C., Schwarz, H. and Mitschang, B. (2014b), "The Manufacturing Knowledge Repository. Consolidating Knowledge to Enable Holistic Process Knowledge Management in Manufacturing", in *Proceedings of the International Conference on Enterprise Information Systems (ICEIS) 2014,* SciTePress, pp. 39–51.

Groover, M.P. (2008), *Automation, production systems, and computer-integrated manufacturing,* 3rd ed., Prentice Hall, Upper Saddle River.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. and Witten, I.H. (2009), "The WEKA Data Mining Software: an Update", *SIGKDD Explorations*, Vol. 11 No. 1, pp. 10–18.

Hjelmervik, O.R. and Wang, K. (2006), "Knowledge Management in Manufacturing: The Soft Side of Knowledge Systems", in Wang, K., Kovacs, G.L., Wozny, M. and Fang, M. (Eds.), *Knowledge Enterprise: Intelligent Strategies in Product Design, Manufacturing, and Management*, Vol. 207, Springer, pp. 89–94.

Holtewert, P., Wutzke, R., Seidelmann, J. and Bauernhansl, T. (2013), "Virtual Fort Knox - Federative, secure and cloud-based platform for manufacturing", in *Proceedings of the CIRP Conference on Manufacturing Systems (CMS) 2013*, Elsevier, pp. 527–532.

Hoos, E., Gröger, C. and Mitschang, B. (2014), "Mobile Apps in Engineering: A Process-Driven Analysis of Business Potentials and Technical Challenges", in *Proceedings of the CIRP Conference on Intelligent Computation in Manufacturing Engineering (CIRP ICME) 2014*, Procedia CIRP Vol. 33, Elsevier, pp. 17–22.

ISA (2000), *Enterprise-Control System Integration* No. ISA-95.

Kassner, L., Gröger, C., Mitschang, B. and Westkämper, E. (2014), "Product Life Cycle Analytics - Next Generation Data Analytics on Structured and Unstructured Data", in *Proceedings of the CIRP Conference on Intelligent Computation in Manufacturing Engineering (CIRP ICME) 2014*, Procedia CIRP Vol. 33, Elsevier, p. 35–40.

Kassner, L. and Mitschang, B. (2016), "Exploring Text Classification for Messy Data: An Industry Use Case for Domain-Specific Analytics", in *Proceedings of the International Conference on Extending Database Technology (EDBT) 2016*, OpenProceedings.org, to appear.

Kemper, H.-G., Baars, H. and Lasi, H. (2013), "An Integrated Business Intelligence Framework. Closing the Gap Between IT Support for Management and for Production", in Rausch, P., Sheta, A.F. and Ayesh, A. (Eds.), *Business Intelligence and Performance Management. Theory, Systems and Industrial Applications*, *Advanced Information and Knowledge Processing*, Springer, London, pp. 13–26.

Königsberger, J., Silcher, S. and Mitschang, B. (2014), "SOA-GovMM: A meta model for a comprehensive SOA governance repository", in *Proceedings of the IEEE International Conference on Information Reuse and Integration (IRI) 2014*, IEEE, pp. 187–194.

MacDougall, W. (2014), "Industrie 4.0 – Smart Manufacturing for the Future", available at: http://www.gtai.de/GTAI/Content/EN/Invest/_SharedDocs/Downloads/GTAI/Brochures/Industries/industrie4.0-smart-manufacturing-for-the-future-en.pdf (accessed 29.10.15).

Meehan, M. (2014), "SOA adoption marked by broad failure and wild success", available at: http://searchsoa.techtarget.com/news/1319609/SOA-adoption-marked-by-broad-failure-and-wild-success (accessed 28.10.15).

Minguez, J., Lucke, D., Jakob, M., Constantinescu, C. and Mitschang, B. (2010), "Introducing SOA into Production Environments - The Manufacturing Service Bus", in *Proceedings of the 43rd CIRP International Conference on Manufacturing Systems (CMS)*, Neuer Wissenschaftlicher Verlag, Wien, pp. 1117–1124.

NHTSA (2014), "NHTSA Data", available at: http://www-odi.nhtsa.dot.gov/downloads/ (accessed 28.10.15).

Ogren, P.V. and Bethard, S.J. (2009), "Building test suites for UIMA components", in *Proceedings of the Workshop on Software Engineering, Testing, and Quality Assurance for Natural Language Processing (SETQA-NLP)*, ACM, pp. 1–4.

Papazoglou, M.P., Heuvel, W.-J.v. and Mascolo, J.E. (2015), "A Reference Architecture and Knowledge-Based Structures for Smart Manufacturing Networks", *IEEE Software*, Vol. 32 No. 3, pp. 61–69.

Sebastian-Coleman, L. (2013), *Measuring data quality for ongoing improvement*, Elsevier, Burlington.

Shi, J., Wan, J., Yan, H. and Suo, H. (2011), "A survey of Cyber-Physical Systems", in *Proceedings of the International Conference on Wireless Communications and Signal Processing (WCSP)*, IEEE, Piscataway, pp. 1–6.

Silcher, S., Dinkelmann, M., Minguez, J. and Mitschang, B. (2013), "Advanced Product Lifecycle Management by Introducing Domain-Specific Service Buses", in Cordeiro, J., Maciaszek, L.A. and Filipe, J. (Eds.), *Enterprise Information Systems (ICEIS) 2013. Revised Selected Papers*, *Lecture Notes in Business Information Processing*, Vol. 141, Springer Berlin, pp. 92–107.

Vogel-Heuser, B., Kegel, G., Bender, K. and Wucherer, K. (2009), "Global information architecture for industrial automation", *Automatisierungstechnische Praxis*, Vol. 51 No. 01-02, pp. 108–115.

Wang, R.Y. and Strong, D.M. (1996), "Beyond accuracy: what data quality means to data consumers", *Journal of Management Information Systems*, Vol. 12 No. 4, pp. 5–33.

Westkämper, E. (2014), *Towards the Re-Industrialization of Europe. A concept for manufacturing for 2030*, Springer, Berlin.

Whitman, M.E. and Mattord, H.J. (2007), *Principles of information security,* 3rd ed., Thomson Course Technology, Boston.

Zuehlke, D. (2010), "SmartFactory - Towards a factory-of-things", *Annual Reviews in Control*, Vol. 34 No. 1, pp. 129–138.

ZVEI (2015), "The Reference Architectural Model Industrie 4.0 (RAMI 4.0)", available at: http://www.zvei.org/Downloads/Automation/ZVEI-Industrie-40-RAMI-40-English.pdf (accessed 28.09.15).

Resources Planning in Database Infrastructures

Eden Dosciatti[1], Marcelo Teixeira[1], Richardson Ribeiro[1], Marco Barbosa[1], Fábio Favarim[1], Fabrício Enembreck[2] and Dieky Adzkiya[3]

[1]*Federal University of Technology-Paraná, Pato Branco, Brazil*

[2]*Pontificial Catholical University-Paraná, Curitiba, Brazil*

[3]*Delft University of Technology, Delft, The Netherlands*

{*edenrd, marceloteixeira, richardsonr, mbarbosa, favarim*}*@utfpr.edu.br, fabricio@ppgia.pucpr.br, d.adzkiya@tudelft.nl*

Keywords: Modeling, Simulation, Resources Planning, Performance, Availability.

Abstract: Anticipating resources consumption is essential to project robust database infrastructures able to support transactions to be processed with certain quality levels. In Database-as-a-Service (DBaaS), for example, it could help to construct Service Level Agreements (SLA) to intermediate service customers and providers. A proper database resources assessment can avoid mistakes when choosing technology, hardware, network, client profiles, etc. However, to be properly evaluated, a database transaction usually requires the physical system to be measured, which can be expensive an time consuming. As most information about resource consumption are useful at design time, before developing the whole system, is essential to have mechanisms that partially open the black box hiding the in-operation system. This motivates the adoption of predictive evaluation models. In this paper, we propose a simulation model that can be used to estimate performance and availability of database transactions at design time, when the system is still being conceived. By not requiring real time inputs to be simulated, the model can provide useful information for resources planning. The accuracy of the model is checked in the context of a SLA composition process, in which database operations are simulated and model estimations are compared to measurements collected from a real database system.

1 INTRODUCTION

Transaction processing is a crucial part of the development of modern web systems, such as those based on *Service-Oriented Architecture* (SOA), a new paradigm to compose distributed business models. In SOA, an entire transaction is usually composed by distinct phases, such as networking, service processing, database processing, third-part processing, etc. For resources planning, it is usual that each particular phase is individually approached. In this paper, we concentrate on evaluating database transaction processing, especially for SOA systems (although not only), complementing previous results focused on the other phases of SOA (Rud et al., 2007; Bruneo et al., 2010; Teixeira et al., 2015).

In SOA, transactions are directly related to *Quality of Service* (QoS), and *Service Level Agreements* (SLAs) are mechanisms used to legally express commitments among service customers and providers (Sturm et al., 2000). Performance and availability of database operations are examples of clauses that can be agreed in SLA, specially when the database itself is provided as a service (DBaaS).

The effects of not being able to fulfill a database SLA are many. This kind of transaction commonly appears in the context of a service composition, as a particular stage of an SOA application. Therefore, if it fails to fulfill the metrics accorded in an SLA, this will probably affect the overall web service behavior and, as a consequence, the overall service orchestration, in a ripple effect, breaching one or more SLAs. Thus, for an entire SOA process, it is important to prevent a database transaction to fail or, at least, to be able to anticipate when it is susceptible to happen.

This task may not be so easy, as the ratio of load variation in web applications can reach the order of 300% (Chase et al., 2001), making it difficult to anticipate QoS. What is observed is that applications are entirely developed to be then stressed and measured, which can be quite expensive and time consuming. Recent works have suggested that SOA QoS can be estimated by modeling (Rud et al., 2007; Bruneo et al., 2010; Teixeira et al., 2015), but they have basically focused on networking and processing stages, assuming that database time consumption is implicit, which may be a strong assumption, as illustrated in (Teixeira and Chaves, 2011).

Dosciatti, E., Teixeira, M., Ribeiro, R., Barbosa, M., Favarim, F., Enembreck, F. and Adzkiya, D.
Resources Planning in Database Infrastructures.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 53-62
ISBN: 978-989-758-187-8

In this paper, we propose a stochastic modeling approach to estimate performance and availability of database transactions susceptible to intense workloads. By adopting *Generalized Stochastic Petri Nets* (GSPNs) as modeling formalism, we construct a formal structure that can be simulated and estimations can be used to anticipate resource consumption of database operations running under different load profiles. Based on these estimations, it is furthermore shown how to construct, at modeling time, realistic contracts for database transactions, which can be naturally combined as part of the estimations provided in works such as in (Rud et al., 2007; Bruneo et al., 2010; Teixeira et al., 2015).

The main advantage of our approach is not requiring real-time measurements nor the complete system implementation to be simulated. These information may not be available at design-time, when resources allocation is conducted. Instead, the model supports high level parameters collected from the *Data Base Management System* (DBMS) and statistics collected from samples of database query execution. For this reason, database technology, infrastructure or particular type of operation to be simulated, are implicit into the simulation scheme.

An example of a contract composition process is presented to illustrate the proposed approach. Using parts of a real database system and samples of relational database operations, we collect the input parameters to the model, which is then simulated and estimations are collected. Afterwards, we validate the estimations. This could be done by comparing them to benchmark data. In this paper, however, we are more interested on the uncertainty observed in the real-time behavior of transactions, e.g., how transactions behave when parameters change, or what is the performance degradation when workload increases, or what is the rate of requests queueing for a load profile, etc. These informations are not directly available from benchmarks, since they focus mostly on best and worst cases, for example. To be possible to check the accuracy of the proposed model so, we compare its estimations to measurements collected from a real database system. Results indicate that it is possible to trace the real behavior keeping a stochastically-reasonable average of 80% accuracy.

The paper is organized as follows: Section 2 discusses the related work; Section 3 introduces the basic concepts of SOA, SLA and GSPN; Section 4 presents the proposed GSPN model. Section 5 presents an example and some final comments are discussed in Section 6.

2 RELATED LITERATURE

Performance of databases has been a concern since the firstly proposed technologies and relational models (Elhardt and Bayer, 1984; Adams, 1985). From the web advent, however, advanced features have been combined to the existent DBMSs, attempting to support emergent requirements such as parallelism, distribution (Dewitt and Gray, 1992), object (Kim et al., 2002) and service-orientation (Tok and Bressan, 2006), etc. Although the interest on new technologies has recently grown, it has become more and more difficult to estimate their behavior.

In particular, when a database is part of a service, or when it is provided as a service itself, it is usually exposed to a highly variable and data-intensive environment, which makes it critical to estimate its QoS levels. In (Ranganathan et al., 1998), it has been discussed the impact of radically different workload levels on the database performance and how it becomes a concern when the database is immersed in QoS-aware frameworks that require QoS guarantees (Lin and Kavi, 2013). In general, the literature tackle this concern using run-time policies to filter and balance the database load (Lumb et al., 2003; Schroeder et al., 2006; Krompass et al., 2008). When connecting business partnerships, however, the negotiation of QoS criteria starts much earlier, at the service design phase, as it is necessary to plan and compose SLA clauses to be agreed.

An option to cover this gap is by adopting analytic models. For example, in (Tomov et al., 2004) it has been proposed a queuing network model to estimate the response time of database transactions. Furthermore, in (Osman and Knottenbelt, 2012) it has been compared the performance of different database designs via modeling. Queue time is predicted by using heuristic rules in (Zhou et al., 1997). Besides not being natively constructed for web environments, this approaches are also predominantly deterministic, which often does not match the characteristics of the real web environments (Teixeira et al., 2011) and can compromise the accuracy when estimating transactions with variable workloads. In addition, they are not usually flexible enough to be quickly converted in practical tools, or to be modified to analyze different system orchestrations, etc.

Thus, the need for supporting database QoS estimation remains. This is a quite challenging task, as web environments practically lack execution patterns and can present highly variable workloads, making it critical for a transaction to be estimated (Nicola and Jarke, 2000). In the same way, it is conceivably difficult to ensure that database queries will execute

quickly enough to keep the process flow, avoiding it to be delayed more than expected (Reiss and Kanungo, 2005). The modeling approach to be presented in this paper is an option to face these challenges and implement database resources planning.

3 RELATED CONCEPTS

SOA comprises a set of principles for software development, fundamentally based on the concept of *service* (Josuttis, 2008). A service is a self-contained component of software that receives a request, processes it, and returns an answer. Eventually, a particular step of a service execution involves to access a database structure an process a data transaction. It may happen that a database transaction is itself offered as a service (DBaaS). In this case, the transaction processing is even more critical, as it is susceptible to a data-intensive environment, and its behavior becomes difficult to be estimated.

In SOA, legal commitments on services, including database transactions, are expressed by a mechanism known as *Service Level Agreements* (SLA) (Sturm et al., 2000). An SLA expresses obligations and rights regarding levels of QoS to be delivered and/or received. SLA clauses usually involve metrics such as response time, availability, cost, etc., and also establish penalties to be applied when a delivered service is below the promised standard (Raibulet and Massarelli, 2008).

In practice, ensuring that a SOA system will behave as expected is very difficult, and so it is difficult to compose, at design time, realistic SLAs. An alternative to probabilistically estimate the behavior of a service is given by modeling approaches. A model enables to observe the service behavior under "pressure", without exactly constructing the whole system.

The model described in this paper serves to this purpose and it is modeled by Petri nets. *Petri net* (PN) (Reisig and Rozenberg, 1998) is a formalism that combines a mathematical foundation to an intuitive modeling interface that allows to model systems characterized by concurrency, synchronization, resources sharing, etc. These features appear quite often in SOA systems, which make PNs a natural modeling choice.

Structurally, a Petri net is composed by *places* (modeling states), *transitions* (modeling state changes), and *directed arcs* (connecting places and transitions). To express the conditions that hold in a given state, places are marked with *tokens*.

Extensions of Petri nets have been developed to include the notion of time (Murata, 1989), which allows to represent time-dependent processes, such as communication channels, code processing, hardware designs, system workflows, etc. *Generalized Stochastic Petri Nets* (GSPNs) (Kartson et al., 1995), for example, is an extension that combines timed and non-timed PNs. In GSPN, *time* is represented by random variable, exponentially distributed, which are associated to *timed transitions*. When, for a given transition, the time is irrelevant, then one can simply use *non-timed* (or *immediate*) transitions.

Formally, a GSPN is a 7-tuple $GSPN = \langle P, \mathcal{T}, \Pi, I, O, M, W \rangle$, where:

- $P = \{p_1, p_2, \ldots, p_n\}$ is a finite set of places;
- $\mathcal{T} = \{t_1, t_2, \ldots, t_m\}$ is a finite set of transitions;
- $\Pi : \mathcal{T} \to \mathbb{N}$ is the priority function, where:

$$\Pi(t) = \begin{cases} \geq 1, & \text{if } t \in \mathcal{T} \text{ is immediate;} \\ 0, & \text{if } t \in \mathcal{T} \text{ is timed.} \end{cases}$$

- $I : (\mathcal{T} \times P) \to \mathbb{N}$ is the input function that defines the multiplicities of directed arcs from places to transitions;
- $O : (\mathcal{T} \times P) \to \mathbb{N}$ is the output function that defines the multiplicities of directed arcs from transitions to places;
- $M : P \to \mathbb{N}$ is the initial marking function. M indicates the number of tokens[1] in each place, i.e., it defines the state of a GSPN model;
- $W : \mathcal{T} \to \mathbb{R}^+$ is the weight function that represents either the immediate transitions weights (w_t) or the timed transitions rates (λ_t), where:

$$W(t) = \begin{cases} w_t \geq 0, & \text{if } t \in \mathcal{T} \text{ is immediate;} \\ \lambda_t > 0, & \text{if } t \in \mathcal{T} \text{ is timed.} \end{cases}$$

The relationship between places and transitions is established by the sets ${}^\bullet t$ and $t^\bullet$, defined as follows.

Definition 1. *Given a transition $t \in \mathcal{T}$, define:*

- ${}^\bullet t = \{p \in P \mid I(t,p) > 0\}$ *as the pre-conditions of t;*
- $t^\bullet = \{p \in P \mid O(t,p) > 0\}$ *as the post-conditions of t.*

A state of a GSPN changes when an enabled transition fires. Only enabled transitions can fire. *Immediate* transitions fire as soon as they get enabled. The *enabling rule* for firing and the *firing semantics* are defined in the sequel.

Definition 2 (Enabling Rule)**.** *A transition $t \in \mathcal{T}$ is said to be enabled in a marking M if and only if:*

- $\forall p \in {}^\bullet t, M(p) \geq I(t,p)$.

[1] Black dots are usually used to graphically represent a token in a place.

When an enabled transition fires, it removes tokens from input to output places (its *pre* and *post* conditions).

Definition 3 (Firing Rule). *The firing of transition $t \in \mathcal{T}$ enabled in the marking M leads to a new marking M' such that* $\forall p \in ({}^\bullet t \cup t^\bullet), M'(p) = M(p) - I(t,p) + O(t,p)$.

A GSPN is said to be *bounded* if there exists a limit $k > 0$ for the number of tokens in every place. Then, one ensures that the state-space resulting from a bounded GSPN is finite.

When the number of tokens in each input place p of t is N times the minimum needed to enable t ($\forall p \in {}^\bullet t, M(p) \geq N \times I(t,p)$, where $N \in \mathbb{N}$ and $N > 1$), it enables the transition to fire more than once. In this situation, the transition t is said to be enabled with degree $N > 0$. Transition firing may use one of the following dynamic semantics:

- *single-server*: N sequential fires;
- *infinite-server*: N parallel fires;
- *k-server*: the transition is enabled up to k times in parallel; tokens that enable the transition to a degree higher than k are handled after the first k firings.

It can be shown (Kartson et al., 1995; Marsan et al., 1984) that GSPNs are isomorphic to *Continuous-Time Markov Chains* (CTMC). However, it is more expressive, as it allows to compute metrics by both simulation and analysis of the state space. In the last case, GSPN are indeed converted into CTMC for analysis. Furthermore, GSPNs allow to combine exponential arranges to model different time distributions (Desrochers, 1994), which is useful to capture specific dynamics of systems.

4 PROPOSED MODEL

The modeling proposed in this paper starts when a given web service requests a database operation. When received in the DBMS, this request is buffered, processed and buffered again, when an answer is ready to be replied back to requestor. When this happens, our modeling finishes. For this scenario, we model the subphases of a database transaction in GSPN: *Buffering* and *Processing*, as shown in Fig. 1.

Table 1 summarizes the model's notation.

Buffering Structure: The model firstly runs when the timed transition T_λ fires tokens toward the place B_I. Fired tokens model database requests and B_I models the DBMS buffer. The firing rate is defined by $1/d_\lambda$, where d_λ is the delay assigned to T_λ. The limit

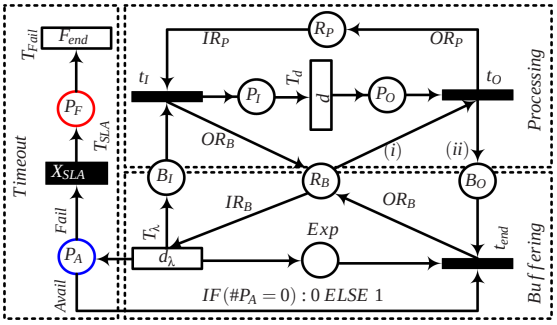


Figure 1: GSPN model.

Table 1: Notation of the GSPN model.

Places	
Exp	expectation of tokens to be processed;
R_B	resources available for buffering;
B_I	input buffer;
B_O	output buffer;
R_P	resources available for processing;
P_I	requests stored before processing;
P_O	requests stored after processing;
P_A	requests successfully attended;
P_F	requests that have failed.
Transitions	
T_λ	requests arrivals (delay d_λ);
T_d	requests processing (delay d);
T_{SLA}	requests failing (delay X_{SLA});
t_I	processing Input;
t_O	processing Output;
t_{end}	process exit point;
T_{Fail}	timeout exit point.

of tokens to be received in B_I is controlled by the number of tokens available in the place R_B, which are also shared with the output buffer B_O. In order to count the expectation of tokens into the model, and consequently to be able to estimate their performance, we create a place named Exp, that receives a copy of each token arriving in the system, and loses a token whenever the transition t_{end} fires.

Processing Structure: From B_I, tokens are moved to the place P_I, which models the processing phase. The place R_P controls the number of requests that can be concurrently processed. Tokens remain in P_I as long as it takes for them to be processed, which is modeled by the delay d of the transition T_d. After processed, T_d fires moving tokens to P_O from where the immediate transition t_O transfers them to the output buffer B_O. Remark that tokens leave the processing phase if and only if there exist enough resources in R_B. On the contrary they remain in P_O, waiting for buffering resources. From B_O, tokens immediately leave the model (by t_{end}), which represents the requestor being answered.

Timeout Structure: When the transition T_λ firstly fires, besides to send a token to B_I (performance model), it also copies it in the place P_A. The idea is to be able to estimate how many requests delay longer than a predefined response time. For that purpose, we assign to X_{SLA} the time we intend to wait until counting a failure. If the performance model reaches t_{end} first, P_A loses the token and the transaction is successfully completed. If X_{SLA} fires first, the transaction is also completed (because the arc *Avail* gets 0), but a failure is registered, i.e., a token reaches P_F.

Repository of Resources: Two repository comprise our model: R_B (buffering resources) and R_P (processing resources). From/to R_B and R_P, we connect arcs representing the number of tokens simultaneously moved when a source transition fires. We denote by IR_B and IR_P the resources consumption and by OR_B and OR_P the resources refunding from/to R_B and R_P, respectively. We assume that the number of tokens moved from/to the repositories is conservative.

Blocking: By sharing R_B with two consumers, B_I and B_O, we actually design a possibly blocking model. In fact if B_I consumes all resources in R_B, then tokens cannot leave the processing phase. At the same time, T_λ cannot fire any more tokens to B_I and, so, the model is deadlocked. We avoid this by assigning two logical conditions (*(i)* and *(ii)*) to the arcs that lead to the place B_O, where:

$$(i):\ IF\ \ (\#R_B < IR_B) : 0\ \ ELSE\ \ IR_B;$$

$$(ii) : IF\ \ (\#R_B < IR_B) : 0\ \ ELSE\ \ 1.$$

The formulas (i) and (ii) are syntactically compliant to the *TimeNET* tool, adopted in this paper. Essentially, the condition (i) avoids the deadlock by firing t_O even without enough resources in R_B. When this is the case, the condition (ii) assigns 0 to the arc that leads to B_O and the token leaves the system. In practice, this models a situation when the DBMS rejects new transactions while the system is completely full, but as soon as any request is processed, transactions get to be received again.

4.1 Model Parameters

To be simulated, the GSPN model requires to be set up with parameters that connect it to the behavior of the system that has been modeled. We show in the following how such parameters can be derived.

4.1.1 Buffering Parameters

We first define a marking[2] for R_B, i.e., the number of resources available for buffering. This is defined according to the real buffer size, measured in the DBMS. Remark that each DBMS defines a particular amount of memory to be used for database operations and this can be tuned. The parameters we have to collect from the DBMS are:

- *Memory Pages* ($\mathcal{M}^P$): number of blocks of memory allocated for database operations;
- *Memory Page Size* ($\mathcal{M}_s^P$): amount of bytes assigned to each $\mathcal{M}^P$.

Remark that the greater the number of memory pages, the faster is the transfer from disk to memory, but the greater is rate of I/O communication, which is usually time expensive. On the other hand, the larger the memory page, the slower the transfer to memory.

As from

$$\mathrm{Av}^{\mathcal{M}} = \mathcal{M}^P \cdot \mathcal{M}_s^P$$

we have the amount of memory available to store messages from/to the database system, then the marking of R_B is such that

$$\#R_B = \mathrm{Av}^{\mathcal{M}}.$$

Once R_B is marked, we model its resources consumption by assigning weights to the arcs IR_B and OR_B. To define those values, we have to collect the mean size (bytes) of:

- Ω^{In}: messages received in the database system;
- Ω^{Out}: messages produced by the system as answer.

Thus, $IR_B = \Omega^{\mathrm{In}}$ and $OR_B = \Omega^{\mathrm{Out}}$. Ω^{In} and Ω^{Out} can be derived from samples of database transactions.

After assigning $\#R_B$, IR_B and OR_B to the GSPN, it becomes already possible to estimate the database *Buffering Response Time* (B^{RT}), taking into account the concept of *Mean Response Time* (M^{RT}). In Petri net, M^{RT} results from the *expectation* (ξ) of marking in a given place X ($\xi(X)$), with respect to: (i) the rate (λ) of requests; or (ii) the delay (d) between requests, i.e.,

$$(i)\ M^{RT} = \frac{\xi(X)}{\lambda} \text{ or, equivalently,} (ii)\ M^{RT} = \xi(X) \cdot d.$$

Tools like *TimeNET* syntactically implement these formulas respectively by

$$(i)\ M^{RT} = \xi(X)/\lambda \quad \text{and} \quad (ii)\ M^{RT} = E\{\#X\} \cdot d.$$

So, B^{RT} can be estimated as follows:

$$B^{RT} = \frac{\xi(B_I) + \xi(B_O)}{\lambda}.$$

Note that λ simply results from $1/d_\lambda$, where d_λ is the delay of the timed transition T_λ. In practice, B^{RT} represents the average of time spent by transactions before and after processing.

[2] "#" denotes the marking of a place p, for $\#p \in \mathbb{N}$.

4.1.2 Processing Parameters

This phase starts when t_I fires tokens towards the place P_I, finishing when the transition t_O releases them. There are basically four processing parameters to be derived: the delay d for the transition T_d, the marking for R_P and the weights for the arcs IR_P and OR_P, which connect the model from/to the repository R_P of processing resources.

Marking R_P requires to measure the system in order to collect the major number of operations simultaneously supported by the DBMS, without queueing requests. This can be done by gradually increasing the workload of requests until the point where the system starts to queue. This specific point can be detected by a sudden increase in the response time, when the processing resources are at all consumed.

Thus, $\#R_P$ receives the value of the workload applied before observing evidences of queue, and 1 is assigned to the weight of the arcs IR_P and OR_P.

Processing Response Time (P^{RT}):

$$P^{RT} = \frac{\xi(P_I) + \xi(P_O)}{\lambda},$$

where, $\xi(P_I)$ and $\xi(P_O)$ are respectively the expectation of marking in P_I and P_O.

4.1.3 Database Mean Response Time

From B^{RT} and P^{RT}, one can estimate the *overall database* M^{RT} by:

$$\Sigma^{RT} = \frac{\xi(Exp)}{\lambda} \quad \text{or, equivalently,} \quad \Sigma^{RT} = B^{RT} + P^{RT}.$$

5 MODEL ASSESSMENT

Consider the process shown in Fig. 2.

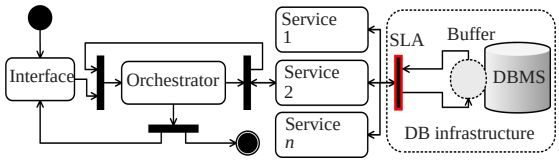


Figure 2: Evaluated Process.

The process starts when remote users invoke an orchestration service, via a web browser. Requests are organized according to the process workflow, and prepared to access remote services, which may access other services or interact with databases (dashed circle). Between a service and its consumer, a SLA regulates the QoS that is to be offered. Usually, this SLA is empirically constructed and, as a consequence, it is not rare to observe services delaying longer than the minimum necessary to match their contracts, which can entail legal penalties for providers, bad reputation for services, money loses for customers, and so on. Our goal here is to anticipate the behavior of the database service when it is variably accessed.

5.1 Database Construction

For the experiments that follow, we consider a partial structure of a relational database system, composed by the following structures:

- `PRODUCT` (ProdID, ProdDesc, ProdColor)
- `CLIENT` (CliID, CliName, CliAddress)
- `INVOICE` (InvID, InvDate, InvValue, ShipmentDate, DeadlineDate, FKClient$^{\#}$)
  - FKClient references CLIENT
- `MOVINVOICE` (Quant, Discount, UnitValue, Label, Status, FKInvoice$^{\#}$, FKProduct$^{\#}$)
  - FKInvoice references INVOICE,
  - FKProduct references PRODUCT

In order to access the database, we implement the following operations in *Relational Algebra*[3].

$$\begin{aligned} C &\leftarrow \Pi^*(Client) \\ I &\leftarrow \Pi^*(Invoice) \\ M &\leftarrow \Pi^*(MovInvoice) \\ P &\leftarrow \Pi^*(Product) \end{aligned}$$

Define query 1:

$$\begin{aligned} \Pi^*\,(\quad & \sigma\,(I.ShipmentDate \quad <= '10/08/2015' \\ & \wedge I.DeadlineDate \quad <= '11/05/2015')) \\ & (C \bowtie I \bowtie M \bowtie P) \end{aligned}$$

Define query 2:

$$\begin{aligned} \Delta \quad \leftarrow \quad & \Pi^*\,(\sigma\,(M.UnitValue >= 5.000{,}00 \\ & \wedge \quad M.FKProduct = 23))\,(M) \end{aligned}$$

$$\begin{aligned} M \quad \leftarrow \quad & \Pi\, Quant, Discount, UnitValue, Status, \\ & Label \leftarrow 'Profitable'\,(\Delta) \end{aligned}$$

Define query 3:

$$\begin{aligned} \Lambda \quad \leftarrow \quad & \Pi\, I.InvID, I.InvValue, P.ProdID, 'Delayed', 'Sold' \\ & (\sigma \quad (I.DeadlineDate <= 'CurrentDate' \\ & \wedge \quad I.ShipmentDate >= '10/10/2015' \\ & \wedge \quad I.InvValue >= 100.000{,}00)) \\ & (I \bowtie M \bowtie P) \\ M \quad \leftarrow \quad & M \cup \{Quant, Discount, \Lambda\}. \end{aligned}$$

[3]Notation * refers to all attributes from a relation.

Query 1 returns the clients and their respective invoices, admitting that: (i) the products had already been shipped; (ii) the deadline for payment will be in at most a month. Query 2 updates the status of a financial transaction (Relation `MOVINVOICE`), labeling it as *profitable* if a given price matches. Finally, Query 3 inserts into a relation results brought from another relation, in a nested instruction.

For simulation, we have considered the respective query versions in *Structured Query Language* (SQL). Optimization and relevance have not been considered when implementing these queries, as we are actually more interested on their timed behavior.

5.2 Database Measurements

Now, we feed our GSPN model. We use an *Apache* tool called *JMeter* (Apache, 2014) to build a test plan that repeatedly executes each query. Then, we gradually increase the workload of requests to observe the point when queues start to appear. That is the point when input parameters are collected. Table 2 presents the inputs to our GSPN model.

Table 2: GSPN Input Parameters.

GSPN Input	Query 1	Query 2	Query 3
Buffering parameters			
$\#R_B$	$\mathcal{M}^P * \mathcal{M}_s^P = 1000 \cdot 4096$		
$IR_B \wedge OR_B$	1435	12	142
Processing parameters			
$\#R_P$	2	3	1
$IR_P \wedge OR_P$	1	1	1
d	286	215	122

Buffering parameters assign values for $\#R_B$, IR_B and OR_B. The marking of R_B is defined according to the DBMS configurations for $\mathcal{M}^P$ and $\mathcal{M}_s^P$. The impact of each operation when allocating resources from R_B, is modeled by the conservative weight of the arcs IR_B and OR_B. By definition, IR_B and OR_B are the measured input and output message sizes, Ω^{In} and Ω^{Out}, respectively.

Processing parameters assign values for $\#R_P$, IR_P, OR_P and d. The marking of R_P models how many instances of a given transaction is supported by the database server. Then, IR_P and OR_P model the impact of each transaction on $\#R_P$, and d represents the mean time required to simultaneously process $\#R_P$ (with no queue formation).

Remark that d represents the probability function that bridges the modeled behavior to the structure that stochastically represents this behavior. Therefore, the value to be assigned to d is obtained by measuring the M^{RT} of samples running in the real system. The number of samples to be considered has to be statistically relevant, usually evidencing a tendency for a stationary behavior. Remark also that every different query to be evaluated may lead to a different value for d and, therefore, has to be individually measured.

5.3 Contract Compositions

Now we exemplify our approach in the context of three challenging questions that are usually faced by engineers when composing SLA contracts. Then, we simulate the model to answer them.

5.3.1 Response Time

Consider the following service contract:

Contract 1: Let $W = \{w_1, w_2, \cdots, w_n\}$ be a set of workloads (requests per second - req/s) possibly arriving at a given DBMS. Which contract for mean response time (M^{RT}) could be guaranteed for w_i, $i = 1, \cdots, n$? As workload variation is quite common over a database structure, whenever w_i changes it becomes more and more difficult to predict the M^{RT} of a transaction, as the system gets to behave nondeterministically, buffering and releasing requests, consuming parallel resources, etc. This makes the rate of performance degradation and recovery unpredictable a priori. However, independently of this variable environment, a service provider is required to deliver his services with M^{RT} no less than the promised standard. Then, it is valuable to know, for each w_i, how many req/s the application supports before exceeding its contract.

We use our model to find out this information. After feeding the model with the statistical data in Table 2, we simulate it for each $w_i \in W$, applied over each proposed query. For the sake of clarity, we cluster our evaluations in three classes of workloads: w_{Light}, w_{Mid} and w_{Heavy}, meaning respectively 1, 5 and 10 req/s. *TimeNet* tool (Zimmermann, 2014) has been used to perform the simulations, considering a confidence level of 95% and a relative error of 10%. In order to check the accuracy of our estimations, we compare the estimated M^{RT} to the M^{RT} measured from the real database system, using the same workload levels. The results are presented in Table 3.

Table 3: Performance evaluation.

Query	*Source*	MRT under w_i			
		w_{Light}	w_{Mid}	w_{Heavy}	$\equiv$
1	System	260	623	1895	
Select	Model	329	405	1989	81%
2	System	278	640	1475	
Update	Model	218	482	1661	73%
3	System	177	815	1995	
Insert	Model	210	646	2193	92%

The accuracies of our estimations are respectively on the order of 81%, 73% and 92%, reaching 82% in a general case, which certainly is reasonable from a stochastic point of view.

For query 1, for example, we have estimated a M^{RT} of 329 *ms*, when simulating with w_{Light}, while the measured M^{RT} has been of 260 *ms*. When increasing the workload to w_{mid}, it has been estimated a M^{RT} of 405 *ms* against the measured 623 *ms*. With w_{Heavy}, we estimate that a transaction takes 1989 *ms* to answer, while the real transaction has taken 1895 *ms*.

As it can be seen, when we increase w_i, the system becomes less deterministic due to presence of queues. Nevertheless, the estimated M^{RT} keeps tracking the real system behavior.

Using our estimations, one can construct realistic contract clauses for services. Two examples are introduced next.

- Suppose, for example, that a service is required to be delivered in at most 700 *ms*. In this case, the model suggests that keeping the system under this contract requires to admit at most a w_{Mid} workload of requests.
- Now, suppose that we know the mean rate of requests arriving in the system. Consider that w_{Heavy} is expected. In this case, the construction of a contract for the M^{RT} would be quite easy. For example, for query 1 it could be defined a M^{RT} contract of 2000 *ms*; for query 2 the M^{RT} contract would be 1700 *ms*; and, for query 3 the M^{RT} contract would be 2200 *ms*.

5.3.2 Contracts with Acceptable Violations

Now, consider the following service contract:

Contract 2: For a given workload level w_i, which agreement for the M^{RT} could be guaranteed, in a way to admit at most 10% *of contract violation* ?

Now, instead of purely estimating the M^{RT}, we derive a refined version of it, admitting a certain percentage of contract violation. This may be a common clause to be defined by lawyers, but this is a quite complex decision for engineers. We show next how to estimate contract 2 by combining our performance and availability models.

For each workload level w_i, $i = Light, Mid, Heavy$, we gradually increase the M^{RT} assigned to the transition T_{SLA} of our availability model. Intuitively, by increasing the acceptable M^{RT} we decrease the failure rate. Table 4 presents the estimations w have obtained for query 1. A similar proceeding can be naturally adopted for the others.

In the second row, we present a range of possible SLA for the M^{RT}. Then we individually assign each M^{RT} to the delay X_{SLA} of our availability structure. Afterwards, we simulate the model, variating w_i for each configuration, collecting the percentage of failure as an answer.

Table 4: Failure evaluation for query 1.

w_i	Suggested SLA for the M^{RT} (*ms*)						
	100	200	300	400	500	600	700
	Estimated failure rate (%)						
w_H	67	45	33	22	17	12	**10**
w_M	52	32	25	18	14	**9**	**7**
w_L	43	24	21	14	**10**	**8**	**6**

For example, by using the workload w_{Light}, we have estimated (Table 3) a M^{RT} of 329 *ms*. Nevertheless, one can observe in Table 4 that 500 *ms* is the minimum M^{RT} that ensures a failure rate of at most 10%. For w_{Mid}, equivalent condition is reached using a M^{RT} of 600 *ms*, while w_{Heavy} requires at least 700 *ms* to satisfy the contract 2.

5.3.3 Contracts with Acceptable Unavailability

Now, consider the following service contract:

Contract 3: Given a prefixed agreement for the M^{RT}, which is the highest workload supported by the system such that the contract is not violated more than 10%?

Contract 3 inversely approaches the problem with respect to contracts 1 and 2. It supposes that the service will be delivered in at most M^{RT}, and the aim is to discover which workload could break this rule. Moreover, it considers to accept a failure rate of at most 10%.

Once again we use query 1 to illustrate the contract 3. We firstly show the contract options for M^{RT}. As query 1 takes 286 *ms* to answer under minimum (Table 2), then we start our simulations by considering a M^{RT} of 300 *ms*. Afterwards, we increase this parameter for eight more scenarios and the results are shown in Table 5.

Table 5: Workload evaluation for query 1.

SLA	Estimated Workload	Failure Rate
M^{RT} *(ms)*	*(Req/sec)*	*($\leq$ 10%)*
300	**0**,**91**	10,00%
400	**1**,**11**	9,98%
500	**1**,**43**	8,68%
600	**1**,**92**	9,99%
700	**2**,**12**	9,99%
800	**4**,**76**	9,86%
900	**10**,**53**	9,89%
1000	**11**,**76**	8,97%

Consider, for example, that a service has to be delivered in at most 700 *ms*. In this case, we inform to the service supplier that his system can support, at

most $2,12$ req/sec and, under this workload, the rate of failure would be stochastically less than 10%.

6 FINAL COMMENTS

In this paper, it has been presented a model to analyze resources allocation in databases infrastructures. The model allows to orchestrate and estimate the performance of a range scenarios, upon different workload profiles. Estimations can then be used as a tool to construct dataafdabase service contracts, besides to be useful for load balancing and scaling in database infrastructures, specially in service-oriented environments.

The approach is illustrated by an example where the performance of database operations is estimated. A comparison against measurements collected from the real database system is conducted to validate the results. The general accuracy of the estimations has been on the order of 80%.

In spite of encouraging results, some challenges remain in the database contracts composition. For example, it is still difficult to identify, among all database requests, those delaying longer than acceptable, which could be helpful to identify advanced classes of contracts. Moreover, we intend to adapt our approach to the optimizer-level, where concurrency could be taken into account. Cache effect analysis is another topic that compose our prospects of future research.

REFERENCES

Adams, E. J. (1985). Workload models for DBMS performance evaluation. In *Proceedings of the 1985 ACM thirteenth annual conference on Computer Science*, CSC '85, pages 185–195, New York, NY, USA. ACM.

Apache (2014). *jMeter 2.3.2*. http://jmeter.apache.org/.

Bruneo, D., Distefano, S., Longo, F., and Scarpa, M. (2010). Qos assessment of ws-bpel processes through non-markovian stochastic petri nets. In *IEEE International Symposium on Parallel Distributed Processing*, pages 1 –12.

Chase, J. S., Anderson, D. C., Thakar, P. N., Vahdat, A. M., and Doyle, R. P. (2001). Managing energy and server resources in hosting centers. In *Symposium on Operating Systems Principles*, Alberta, Canada.

Desrochers, A. A. (1994). Applications of Petri nets in manufacturing systems: Modeling, control and performance analysis. IEEE Press.

Dewitt, D. J. and Gray, J. (1992). Parallel database systems: the future of high performance database systems. *Communications of the ACM*, 35:85–98.

Elhardt, K. and Bayer, R. (1984). A database cache for high performance and fast restart in database systems. *ACM Transactions on Database Systems*, 9:503–525.

Josuttis, N. (2008). *SOA in Practice*. O'Reilly, 1 edition.

Kartson, D., Balbo, G., Donatelli, S., Franceschinis, G., and Conte, G. (1995). *Modelling with Generalized Stochastic Petri Nets*. John Wiley & Sons, Inc., 1st edition.

Kim, S., Son, S., and Stankovic, J. (2002). Performance evaluation on a real-time database. In *Real-Time and Embedded Technology and Applications Symposium, 2002. Proceedings. Eighth IEEE*, pages 253–265.

Krompass, S., Scholz, A., Albutiu, M.-C., Kuno, H. A., Wiener, J. L., Dayal, U., and Kemper, A. (2008). Quality of service-enabled management of database workloads. *IEEE Data(base) Engineering Bulletin*, 31:20–27.

Lin, C. and Kavi, K. (2013). A QoS-aware BPEL framework for service selection and composition using QoS properties. *Int. Journal On Advances in Software*, 6:56–68.

Lumb, C. R., Merchant, A., and Alvarez, G. A. (2003). Façade: Virtual storage devices with performance guarantees. In *Proceedings of the 2nd USENIX Conference on File and Storage Technologies*, pages 131–144, Berkeley, CA, USA. USENIX Association.

Marsan, M. A., Balbo, G., and Conte, G. (1984). A class of generalized stochastic Petri nets for the performance analysis of multiprocessor systems. In *ACM Transactions on Computer Systems*, volume 2, pages 1–11.

Murata, T. (1989). Petri nets: Properties, analysis and applications. *Proceedings of the IEEE, v.77*, pages 541–580.

Nicola, M. and Jarke, M. (2000). Performance modeling of distributed and replicated databases. *IEEE Trans. on Knowl. and Data Eng.*, 12(4):645–672.

Osman, R. and Knottenbelt, W. J. (2012). Database system performance evaluation models: A survey. *Performance Evaluation*, 69(10):471 – 493.

Raibulet, C. and Massarelli, M. (2008). Managing non-functional aspects in SOA through SLA. In *Int. Conference on Database and Expert Systems Application*, Turin, Italy.

Ranganathan, P., Gharachorloo, K., Adve, S. V., and Barroso, L. A. (1998). Performance of database workloads on shared-memory systems with out-of-order processors. *Operating Systems Review*, 32:307–318.

Reisig, W. and Rozenberg, G. (1998). Informal introduction to petri nets. In *Lectures on Petri Nets I: Basic Models*, pages 1–11, London, UK. Springer-Verlag.

Reiss, F. R. and Kanungo, T. (2005). Satisfying database service level agreements while minimizing cost through storage QoS. In *Proceedings of the IEEE International Conference on Services Computing*, volume 2, pages 13–21, Washington, USA.

Rud, D., Schmietendorf, A., and Dumke, R. (2007). Performance annotated business processes in service-oriented architectures. *International Journal of Simulation: Systems, Science & Technology*, 8(3):61–71.

Schroeder, B., Harchol-Balter, M., Iyengar, A., and Nahum, E. (2006). Achieving class-based QoS for transactional workloads. In *Proceedings of the 22nd International Conference on Data Engineering*, Washington, DC, USA. IEEE Computer Society.

Sturm, R., Morris, W., and Jander, M. (2000). *Foundations of Service Level Management*. Sams Publishing.

Teixeira, M. and Chaves, P. S. (2011). Planning database service level agreements through stochastic petri nets. In *Brazilian Symposium on Databases*, Florianopolis, Brazil.

Teixeira, M., Lima, R., Oliveira, C., and Maciel, P. (2011). Planning service agreements in SOA-based systems through stochastic models. In *ACM Symposium On Applied Computing*, TaiChung, Taiwan.

Teixeira, M., Ribeiro, R., Oliveira, C., and Massa, R. (2015). A quality-driven approach for resources planning in service-oriented architectures. *Expert Systems with Applications*, 42(12):5366 – 5379.

Tok, W. H. and Bressan, S. (2006). DBNet: A service-oriented database architecture. *International Workshop on Database and Expert Systems Applications*, pages 727–731.

Tomov, N., Dempster, E., Williams, M. H., Burger, A., Taylor, H., King, P. J. B., and Broughton, P. (2004). Analytical response time estimation in parallel relational database systems. *Parallel Comput.*, 30:249–283.

Zhou, S., Tomov, N., Williams, M. H., Burger, A., and Taylor, H. (1997). Cache modeling in a performance evaluator for parallel database systems. In *MASCOTS*, pages 46–50.

Zimmermann, A. (2014). *TimeNET 4.0*. Technische Universität Ilmenau, URL: http://www.tu-ilmenau.de/TimeNET.

A Graph and Trace Clustering-based Approach for Abstracting Mined Business Process Models

Yaguang Sun and Bernhard Bauer
Software Methodologies for Distributed Systems, University of Augsburg, Augsburg, Germany
{*yaguang.sun, bernhard.bauer*}*@informatik.uni-augsburg.de*

Keywords: Business Process Model Abstraction, Business Process Mining, Workflow Discovery, Graph Clustering, Trace Clustering.

Abstract: Process model discovery is a significant research topic in the business process mining area. However, existing workflow discovery techniques run into a stone wall while dealing with event logs generated from highly flexible environments because the raw models mined from such logs often suffer from the problem of inaccuracy and high complexity. In this paper, we propose a new process model abstraction technique for solving this problem. The proposed technique is able to optimise the quality of the potential high level model (abstraction model) so that a high-quality abstraction model can be acquired and also considers the quality of the sub-models generated where each sub-model is employed to show the details of its relevant high level activity in the high level model.

1 INTRODUCTION

Business process mining techniques aim at discovering, monitoring and improving real processes by extracting knowledge from event logs recorded by enterprise information systems (van der Aalst et al., 2003). The starting point of these techniques is usually an event log which is a set of cases. A case is an instance of a business process and has an attribute *trace* which is a set of ordered events (each event is an instance of a specific activity). In the event log both cases and events are uniquely marked by *case id* and *event id* respectively (van der Aalst, 2011).

As one of the most important learning tasks in business process mining area, the current process model discovery techniques encounter great challenges in the context of real-life event logs. Such logs that usually contain a tremendous number of trace behaviors (expressed by the activities and their precedence relations in the trace) stem from the business processes executed in highly flexible environments, e.g., healthcare, customer relationship management (CRM) and product development (Weerdt et al., 2013). As a result, "spaghetti-like" business process models are often generated while mining real-life event logs with existing workflow discovery techniques. Such models are often inaccurate (in the process mining area the fitness is utilised to express the accuracy of a mined model which measures the proportion of behaviors in the event log possible according to the model) and difficult to be comprehended because of their high complexity. Accordingly, two main pioneering approaches have been developed in the literature to solve this problem: *trace clustering* technique (Weerdt et al., 2013; Bose and van der Aalst, 2009; Bose and van der Aalst, 2010; Song et al., 2009; Ferreira et al., 2007) and *process model abstraction-based* technique (Bose and van der Aalst, 2009; Baier and Mendling, 2013; Conforti et al., 2014).

Trace clustering techniques divide the raw event log into several sub-logs where each sub-log contains the traces with similar behaviors and helps generate a more accurate and comprehensible sub-model. Generally, these techniques perform well for handling the logs with a moderate amount of trace behaviors. Nevertheless, the limitation of current trace clustering techniques will be revealed while dealing with event logs containing massive trace behaviors. For instance, the event log of a Dutch academic hospital from Business Process Intelligence Contest 2011 (BPIC 2011) contains 624 activities among which a large number of relations are exhibited (the average out-degree for each activity is 6.2564) and most of the classical trace clustering methods can not bring a significant improvement on the mining result for this hospital log (as shown in Section 4).

Process model abstraction-based approaches

Sun, Y. and Bauer, B.
A Graph and Trace Clustering-based Approach for Abstracting Mined Business Process Models.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 63-74
ISBN: 978-989-758-187-8

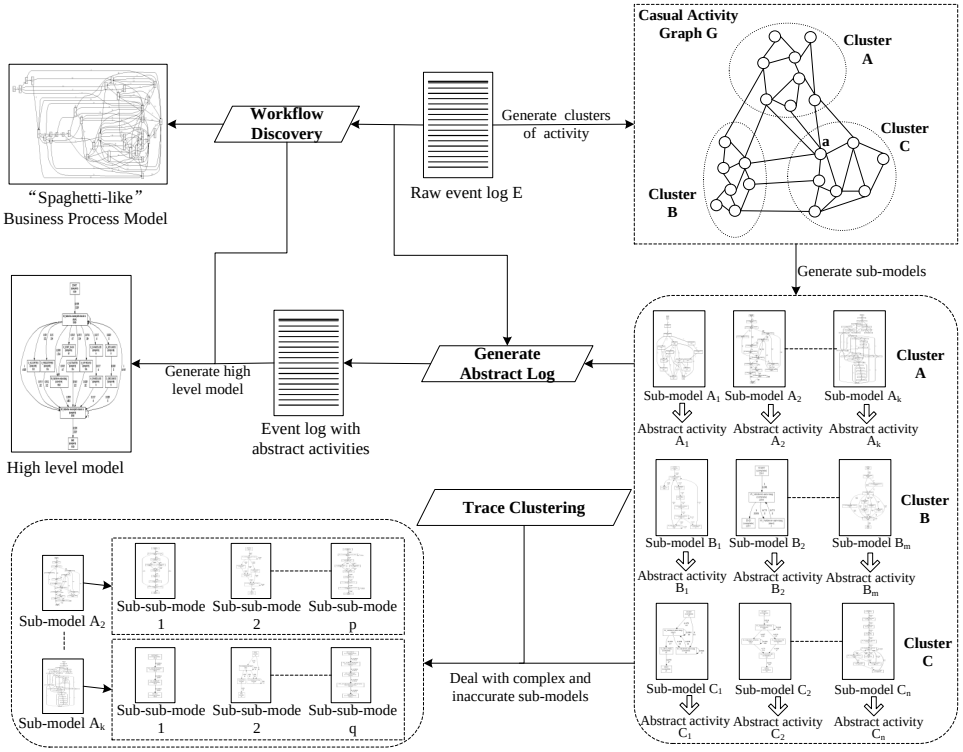


Figure 1: Illustration of the basic ideas of the approach proposed in this paper.

make the assumption that the raw models mined from real-life logs contain low level sub-processes which should be discovered in the form of sub-traces in the original event logs and abstracted into high level activities so that the insignificant low level process behaviors can be hidden in the high level activities. Thus, more accurate and simpler high level process models can be obtained. However, most of the present process model abstraction-based techniques focus mainly on the discovery of sub-processes and can not ensure the accuracy of the high level process models generated.

In this paper, we put forward a new method which inherits the characteristics of the trace clustering techniques and the process model abstraction-based approaches for solving the problem of "spaghetti-like" process models. The proposed technique is able to optimise the quality of the potential high level process model through a new abstraction strategy based on *graph clustering* technique (Schaeffer, 2007). As a result, a high-quality abstraction model can be built. Furthermore, the quality of the sub-models discovered for showing the details of their related high level activities (used for building the final high level model) is also considered by our approach. The structure of the main contents in this paper is organised as:

- A new strategy for abstracting the raw models mined from real-life event logs is discussed in Section 2.
- In Section 3.1, several important concepts that support the method proposed in this paper are reviewed. In Section 3.2, a three-stage model abstraction method based on the strategy proposed in Section 2 is elaborated.
- To test the efficiency of our method, we carry out a case study in Section 4 by applying our approach to three event logs: the repair log from (van der Aalst, 2011), the hospital log from Business Process Intelligence Contest (BPIC) 2011 and the log of the loan and overdraft approvals process from BPIC 2012.

2 BASIC IDEAS

In the real world, seemingly "spaghetti-like" business process models mined from event logs might still have some rules to follow. Sometimes, the main reason for the structurelessness of these mined models is that they contain several extremely complex substructures. However, the relations among these substructures may be straightforward (this is proven in the case study in Section 4). While turning to a specific event log, such kind of phenomenon mentioned

above can be reflected by the existence of several clusters of activities from an event log where the activities in the same cluster are densely connected and the activities in different clusters are sparsely connected (this is also the assumption for our method). For instance, in Figure 1 an event log E contains 22 activities and a *causal activity graph* G can be established by employing the activities from E as vertices and the *casual relations* (Hompes et al., 2014) among these activities as edges. The definitions about *casual activity graph* and *casual relations* of activities are introduced in detail in Section 3.1. According to Figure 1, the vertices in G can be grouped into three clusters by considering the edge structure in such a way that there should be many edges within each cluster and relatively few edges among the clusters.

With the assumption mentioned above, we put forward a new strategy for solving the problem of complex and inaccurate process models mined from real-life event logs. The basic idea is to generate the clusters of activities firstly by following the same rule utilised in the example shown in Figure 1. Afterwards, for each cluster one or several sub-models are generated where each sub-model only contains the activities from its relevant activity cluster. In the example from Figure 1, the sub-models for cluster A are built by using the activities from cluster A. Then, for a complex and inaccurate sub-model, trace clustering technique is employed to split it into several simple and accurate sub-sub-models so that the sub-model can be well comprehended. Finally, these sub-models (not including the sub-sub-models) generated are abstracted into high level activities with which a simple and accurate ultima high level process model is formed. In this paper the high level process model together with the sub-models (each sub-model is related to one high level activity in the high level model built) are used to show the details of the whole business process recorded in event log.

Basically, two major benefits could be acquired from the strategy proposed above. On one hand, the original tough problem (deal with the entire model) met by current trace clustering techniques is transformed into small sub-problems (deal with the sub-models). Specifically, the raw mined model from event log may contain too many behaviors which might be far beyond the abilities of existing trace clustering techniques. However, by distributing the huge amount of behaviors from the original mined model to several small sub-models (each sub-model contains less behaviors but still might be complex and inaccurate) the trace clustering techniques can provide better results while being applied on these sub-models. On the other hand, the number of activity relations among the clusters is kept as small as possible (which means the relations among the high level activities created are kept as few as possible). As a result, the quality of the potential high level process model is optimised to a large extent because it contains a limited number of behaviors among its activities.

3 APPROACH DESIGN

In this section, we propose a new approach that utilises the strategy introduced in Section 2 for solving the problem of "spaghetti-like" process models mined from event logs. In Section 3.1, several important basic concepts and notations related to our technique are discussed. In Section 3.2, the details of our technique are elaborated.

3.1 Preliminaries

Event logs (van der Aalst, 2011) play the significant part of data sources for various kinds of process mining techniques. The basic concepts related to event logs are conveyed by the following definitions.

Definition 1. *(Case)*
Let C be the set of cases. A case $c \in C$ is defined as a tuple $c = (N_c, \Theta_c)$, where $N_c = \{n_1, n_2, \ldots, n_k\}$ is the set of names of case attributes, $\Theta_c : N_c \rightarrow A_c$ is an attribute-transition function which maps the name of an attribute into the value of this attribute, where A_c is the set of attribute values for case c.

A case is an instance of a specific business process and uniquely identified by *case id*. Each case may have several attributes such as trace, originator, timestamp and cost, etc. As one of the most important case attributes, the *trace* of a case is defined as:

Definition 2. *(Trace)*
Let AT be the set of activities, EV be the set of events and each event $ev \in EV$ is an instance of a particular activity $at \in AT$. A trace is a sequence of ordered events from EV.

Definition 3. *(Event Log)*
An event log is defined as $E \subseteq C$, for any $c_1, c_2 \in E$ such that $c_1 \neq c_2$.

Take a simple event log $E_1 = [< a,b,c >^{15}, < a,c,b >^{15}, < a,b >^{3}, < a,c >^{5}]$ for example. This log contains 38 cases (only the case attribute *trace* is exhibited) and four kinds of trace[1]. There are totally

[1] A trace and a kind of trace are two different concepts. Each trace belongs to a unique case. A kind of trace contains several traces which have the same sequence of events.

$3 \cdot 15 + 3 \cdot 15 + 2 \cdot 3 + 2 \cdot 5 = 106$ events and three activities (activity a, b and c) in this log.

In (Hompes et al., 2014) the fundamental theory for *activity clustering* is developed. Two important concepts that support this theory are demonstrated: *Causal Activity Relations* and *Causal Activity Graph*. The technique proposed in this paper will use these concepts for generating the clusters of activities from event logs.

Definition 4. *(Direct and Casual Activity Relations)* Let AT be the set of activities of an event log E. Symbol $\succ_E$ represents a direct relation between two activities from AT and symbol $\succeq_E$ represents a causal relation between two activities from AT. Let $a, b \in AT$ be two activities, $\phi \in [-1.0, 1.0]$ be a threshold, $a \succ_E b = true$ if $|a \succ_E b| > 0$, where $|a \succ_E b|$ is the number of times that a is directly followed by b in E. $a \succeq_E b = true$ if $|a \succeq_E b| \geq \phi$, where $|a \succeq_E b| \in [-1.0, 1.0]$ is the value of casual relation between a and b.

In our approach we utilise the *DependencyMeasure* method introduced in (Weijters et al., 2006) for calculating the value of casual relation between any two activities which is defined as:

$$|a \succeq_E b| = \begin{cases} \frac{|a \succ_E b| - |b \succ_E a|}{|a \succ_E b| + |b \succ_E a| + 1} & \text{if } a \neq b \\ \\ \frac{|a \succ_E a|}{|a \succ_E a| + 1} & \text{if } a = b \end{cases} \quad (1)$$

A $|a \succeq_E b|$ value close to 1.0 implies a high possibility that there exists a direct casual relation between a and b while a value close to –1.0 signifies a high possibility that there exists no casual relation between a and b. A value close to 0 means uncertainty. Take two activities a and c from event log E_1 created above as an example, $|a \succ_{E_1} c| = 15 + 5 = 20$, $|c \succ_{E_1} a| = 0$, so $|a \succeq_{E_1} c| = (20 - 0)/(20 + 0 + 1) \approx 0.95$. Let the threshold $\phi = 0.9$, then a casual relation is judged to exist between a and c because $|a \succeq_{E_1} c| > \phi$.

Definition 5. *(Casual Activity Graph)* Let AT be a set of activities from event log E, $\Upsilon(AT)$ denotes the set of casual activity graphs over AT. A causal activity graph $G \in \Upsilon(AT)$ is a tuple $G = (V, L)$ where $V \in AT$ is the set of vertices and $L \in (V \times V)$ is the set of edges. Each edge in G represents a casual relation between two activities.

In our method we employ an existing graph clustering technique (based on energy model) from (Noack, 2007) for mining the casual activity graphs following the rule that the activities in the same cluster should be densely connected and the activities in different clusters should be sparsely connected. The main reason for us to select this graph clustering technique is that it is able to automatically generate a suitable number of clusters of vertices according to the edge structure of a graph and also has a good performance. The basic knowledge related to graph clustering technique is well introduced in (Schaeffer, 2007).

3.2 A Three-step Algorithm

In this section a process model abstraction algorithm that consists of three main stages is put forward. This algorithm applies the strategy mentioned in Section 2 which considers the quality of both the potential high level model and sub-models generated. Let $\Pi : (SE, STH) \longrightarrow SG$ be a casual activity graph building method, where *SE* is the set of event logs, *STH* is the set of values of thresholds for judging casual relations among activities and *SG* is the set of casual activity graphs, $\Gamma : SG \longrightarrow SC$ be the graph clustering algorithm from (Noack, 2007), where *SC* is the set of all sets of activity clusters. The details of our method is described in Algorithm 1.

Algorithm 1: Abstracting the raw models mined (AM).

Input: an event log E, the threshold ϕ for judging the causal relations among activities, the threshold α for judging if a high level activity generated should be removed or not, the threshold β for searching for merging modes, a sub-model complexity threshold τ and a sub-model accuracy threshold χ, a trace number threshold κ, cluster number n.

Let G be a casual activity graph.

Let C_{ac} be a set of activity clusters.

1: $G \leftarrow Null$
2: $C_{ac} \leftarrow Null$
3: $G = \Pi(E, \phi)$ # build the casual activity graph
4: $C_{ac} = \Gamma(G)$ # mine the activity clusters
5: ***Stage 1:*** Find multi-cluster activities and extract sub-logs.
input: E, C_{ac}.
output: a new set of activity clusters *MC–*C_{ac}, a set of sub-logs *SSE*.
6: ***Stage 2:*** Generate high level activities and high level model.
input: *SSE*, E, α, β.
output: a high level model *HL–M*, a set of high level activities *H–SA*, a set of sub-logs *H–SSE*.
7: ***Stage 3:*** Deal with complex and inaccurate sub-models from *H–SSE*.
input: *H–SSE*, τ, χ, κ, n.
output: a set of sub-models *SSM*.

Output: a high level model *HL–M*, a set of sub-models *SSM*.

3.2.1 Find Multi-cluster Activities and Extract Sub-logs

In this subsection we make the assumption that a set of activity clusters $C_{ac} = \{c_1, c_2, \ldots, c_m\}$ for event log E has been acquired by Algorithm 1. Sometimes, an activity $a \in c_k \in C_{ac}$ may also have a lot of casual relations with the activities from other clusters. For instance, in the casual activity graph G from Figure 1, the activity a that pertains to cluster C is also connected to many activities in cluster A. In the graph clustering research area most of the classical methods developed presume that a vertice of a graph only belongs to one specific cluster. The graph clustering algorithm utilised in our approach also has the same assumption. However, it is a normal situation that some activities in a casual activity graph should pertain to more than one clusters according to the edge structure of the graph. Based on this fact, we develop a new concept named *Multi-cluster Activity* which is defined as:

Definition 6. *(Multi-cluster Activity)*
Let $\Phi : SG \longrightarrow SV$ be a graph density calculation schema, where SG is the set of casual activity graphs and SV is the set of values of graph density. Given a set of activity clusters $C_{ac} = \{c_1, c_2, \ldots, c_n\}$, an activity $a \in c_k \in C_{ac}$ is a multi-cluster activity if $\exists c_m \in C_{ac}$ such that $\Phi(G_m^{'}) \geq \Phi(G_m)$, where $G_m = (V_m, L_m)$ represents the casual activity graph built by using the activities from activity cluster c_m and $G_m^{'} = (V_m \cup a, L_m^{'})$ is a new graph generated by adding the activity a in G_m.

Given a graph $G = (V,L)$, $\Phi(G) = |L|/(|V| \times (|V|-1))$, where $|L|$ and $|V|$ stand for the total number of edges and the total number of vertices in graph G respectively. The main reason to use graph density for judging a multi-cluster activity is that densely connected activities are more likely to cause complex process behaviors that can't be expressed by the utilised workflow discovery algorithms (our approach leave these potential complex behaviors to trace clustering techniques). Our method detects all of the multi-cluster activities in C_{ac} and then distributes each of them to the eligible activity clusters in C_{ac} so that a new set of activity clusters $MC{-}C_{ac}$ can be generated. For example, let $C_{ac}^{'} = \{c_1, c_2, c_3\}$ be a set of activity clusters mined from event log $E^{'}$, $c_1 = \{a,b,c\}$, $c_2 = \{d,e\}$ and $c_3 = \{f,g,h\}$, pretend that $\Phi(G_{c_2}) = 0.5$, $\Phi(G_{c_3}) = 0.8$, $\Phi(G_{c_2}^{+}) = 0.63$ and $\Phi(G_{c_3}^{+}) = 0.7$, where G_{c_2} is the casual graph for cluster c_2, G_{c_3} for cluster c_3, $G_{c_2}^{+}$ is the casual graph generated by adding the activity $a \in c_1$ in G_{c_2} and $G_{c_3}^{+}$ generated by adding the activity a in G_{c_3}. According to Definition 6, a is a multi-cluster activity because $\Phi(G_{c_2}^{+}) > \Phi(G_{c_2})$. Afterwards, a new activity cluster $c_2^{'} = \{a,d,e\}$ is generated by adding a in c_2. Activity a should not be added in c_3 because $\Phi(G_{c_3}^{+}) < \Phi(G_{c_3})$. Let's presume that a is the only multi-cluster activity found, then the new set of activity clusters $MC{-}C_{ac}^{'} = \{c_1, c_2^{'}, c_3\}$ can be generated.

An intuitive proof about the benefit for locating the multi-cluster activities is shown in the example in Figure 1. We assume that the activity a in cluster C is a multi-cluster activity corresponding to cluster A. By adding a to cluster A the original casual graph G can be transformed into $G^{'}$ as shown in Figure 2. In $G^{'}$, the interrelations between cluster A and C are further decomposed which helps improve the quality of the potential high level model.

Whereafter, the stage 1 of Algorithm 1 creates a sub-log for each activity cluster in $MC-C_{ac} = \{mc_1, mc_2, \ldots, mc_n\}$. For example, for the activity cluster $mc_k \in MC-C_{ac}$ a new log E_{mc_k} is built which contains all of the sub-traces extracted from the original event log E where each sub-trace only includes the activities from mc_k. For instance, let $MC{-}C_{ac}^{'} = \{\{a,b,v,c,d\}, \{u,v,x,z\}\}$ be a set of activity clusters generated by stage 1 of Algorithm 1 executed on an event log $E^{'} = \{<a,b,c,d,v,x,z>^{80}, <a,c,d,u,v,x,z>^{150}, <a,b,v,c,d,u,v,z>^{200}\}$ (pretend that v is a multi-cluster activity). For the activity cluster $\{a,b,v,c,d\} \in MC{-}C_{ac}^{'}$ a new sub-log $SE_1^{'} = \{<a,b,c,d,v>^{80}, <a,c,d>^{150}, <a,b,v,c,d>^{200}\}$ can be created by extracting all the sub-traces in $E^{'}$ where these sub-traces only contain the activities from $\{a,b,v,c,d\}$. Similarly, the sub-log $SE_2^{'} = \{<v,x,z>^{80}, <u,v,x,z>^{150}, <u,v,z>^{200}\}$ can be generated for activity cluster $\{u,v,x,z\}$.

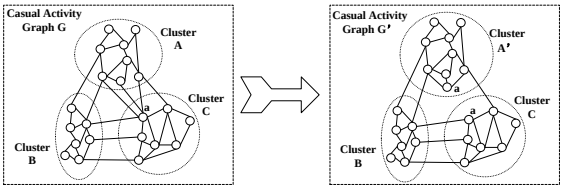


Figure 2: Further decompose the interrelations between cluster A and cluster C.

3.2.2 Generate High Level Activities and High Level Process Model

We presume that the set of sub-logs $SSE = \{E_{mc_1}, E_{mc_2}, \ldots, E_{mc_n}\}$ has been output by the stage 1 of Algorithm 1. Let $\Psi : SE \longrightarrow SS{-}SE$ be a method which splits an event log into several sub-logs where each sub-log contains the traces with the same start

activity and end activity, *SE* represents the set of event logs and *SS−SE* represents the set of all set of sub-logs. Take the simple event log $E^{'} = \{< a,b,c >^{15}, < a,d,c >^{15}, < a,f >^{3}, < a,e,d >^{5}\}$ as an example, $\Psi(E^{'}) = \{E^{'}_1, E^{'}_2, E^{'}_3\}$, where $E^{'}_1 = \{< a,b,c >^{15}, < a,d,c >^{15}\}$, $E^{'}_2 = \{< a,f >^{3}\}$ and $E^{'}_3 = \{< a,e,d >^{5}\}$. The high level activity generation method for the stage 2 of Algorithm 1 is depicted in Algorithm 2.

Algorithm 2: Generate high level activities (GHLA).

Input: the set of sub-logs *SSE*, the original log E, a threshold α, a threshold β.
Let Λ be a trace merging technique which is described in Algorithm 3.
Let *H−SSE* be a set of sub-logs where each sub-log $HE_q \in H{-}SSE$ is relevant to one potential high level activity.
Let *H−SA* be a set of high level activities.
Let *M−SSE* be a set of event logs with merged traces.

1: $H{-}SSE \leftarrow Null$
2: $H{-}SA \leftarrow Null$
3: $M{-}SSE \leftarrow Null$
4: **for** each log $E_{mc_k} \in SSE$ **do**
5: $M{-}SSE \leftarrow M{-}SSE \cup \Lambda(E_{mc_k}, SSE, E, \beta)$
generate log with merged traces.
6: **end for**
7: **for** each log $ME_p \in M{-}SSE$ **do**
8: $H{-}SSE \leftarrow H{-}SSE \cup \Psi(ME_p)$
9: **end for**
10: **for** each log $HE_q \in H{-}SSE$ **do**
11: $H{-}SA \leftarrow H{-}SA \cup HL{-}Activity(q)$
create a high level activity called
$HL{-}Activity(q)$ and put it in $H{-}SA$.
12: **end for**
13: **for** each $HL{-}Activity(p) \in H{-}SA$ **do**
14: **if** $|HL{-}Activity(p)| < \alpha$ **then**
15: remove $HL{-}Activity(p)$ from $H{-}SA$
16: remove HE_p from $H{-}SSE$
$|HL{-}Activity(p)|$ represents the
frequency of occurrence for the
high level activity $HL{-}Activity(p)$.
17: **end if**
18: **end for**

Output: the set of high level activities *H−SA*, the set of sub-logs *H−SSE*.

To explain Algorithm 2 explicitly, an example is employed here (for the rest part of this sub-section). Let $MC{-}C^{'}_{ac} = \{\{a,b,c,d\},\{u,v,x,z\}\}$ be a set of activity clusters generated by stage 1 of Algorithm 1 executed on an event log $E^{'} = \{< a,b,d,u,x,z >^{100}, < a,b,c,d,v,x,z >^{80}, < a,c,d,u,v,x,z >^{150}, < a,b,v,c,d,u,x,z >^{8}\}$, $SSE^{'} = \{E^{'}_{mc_1}, E^{'}_{mc_2}\}$ be a set of sub-logs generated by stage 1 of Algorithm 1 with inputs $MC{-}C^{'}_{ac}$ and $E^{'}$, where sub-log $E^{'}_{mc_1} = \{< a,b,d >^{100}, < a,b,c,d >^{80}, < a,c,d >^{150}, < a,b >^{8}, < c,d >^{8}\}$, sub-log $E^{'}_{mc_2} = \{< u,x,z >^{108}, < v,x,z >^{80}, < u,v,x,z >^{150}, < v >^{8}\}$. A set of sub-logs $H{-}SSE^{'} = \{\{< a,b,d >^{100}, < a,b,c,d >^{80}, < a,c,d >^{150}\}_0, \{< a,b >^{8}\}_1, \{< c,d >^{8}\}_2, \{< u,x,z >^{108}, < u,v,x,z >^{150}\}_3, \{< v,x,z >^{80}\}_4, \{< v >^{8}\}_5\}$ can be generated if $SSE^{'}$ is directly dealt with by the steps $7-9$ of Algorithm 2 (replace the set $M{-}SSE$ in step 7 by using $SSE^{'}$). Afterwards, according to the steps $10-12$ of Algorithm 2 a set of high level activities $H{-}SA^{'} = \{HL{-}Activity(0)^{330}, HL{-}Activity(1)^{8}, HL{-}Activity(2)^{8}, HL-Activity(3)^{258}, HL-Activity(4)^{80}, HL-Activity(5)^{8}\}$ is generated where each high level activity is related to a specific sub-log in $H{-}SSE^{'}$. In our method a high level activity will replace all the sub-traces that exist in its relevant sub-log in $H{-}SSE^{'}$ in the original event log $E^{'}$. For instance, the high level activity $HL{-}Activity(0)$ will replace all the sub-traces from the sub-log $\{< a,b,d >^{100}, < a,b,c,d >^{80}, < a,c,d >^{150}\}_0$ in $E^{'}$. Finally, a high level event log $E^{'}_h = \{< HL-Activity(0), HL-Activity(3) >^{100}, < HL-Activity(0), HL-Activity(4) >^{80}, < HL-Activity(0), HL-Activity(3) >^{150}, < HL-Activity(1), HL-Activity(5), HL-Activity(2), HL-Activity(3) >^{8}\}$ is acquired. The steps $13-18$ of Algorithm 2 remove all the infrequent high level activities generated and their relevant sub-logs in $H{-}SSE^{'}$ either. Removing infrequent activities which is in accordance with the main idea of most advanced process model mining techniques can make the potential model mined concentrate on exhibiting the most frequent process behaviors. In our example, given a threshold $\alpha = 20$, the high level activity $HL{-}Activity(1)$, $HL{-}Activity(2)$ and $HL{-}Activity(5)$ are removed from $H{-}SA^{'}$ and $E^{'}_h$ because the value of their frequency is eight which is smaller than α. At the same time, the sub-logs $\{< a,b >^{8}\}$, $\{< c,d >^{8}\}$ and $\{< v >^{8}\}$ are removed from $H{-}SSE^{'}$. Afterwards, a high level model can be built by mining the generated high level event log $E^{'}_h$ with an existing process model discovery algorithm (this is the way for our method to generate a high level model). Each sub-log in $H{-}SSE^{'}$ will be used to build a sub-model for indicating the details of its relevant high level activity.

Such a design for generating the high level activities will help maintain the precision (van der Aalst,

2011) (precision quantifies the ratio of the behaviors that can be generated by the mined models which are also recorded in the event logs) of the potential high level model together with the sub-models generated compared with the precision of the model mined by using the original log $E^{'}$ (the interested reader can think about it more deeply). Furthermore, our method might generate a huge amount of high level activities while encountering the event logs that have casual graphs with uniform structures. So we make the assumption that the casual graphs of the event logs processed by our method have structures with natural clusters.

Three infrequent high level activities ($HL-Activity(1)$, $HL-Activity(2)$ and $HL-Activity(5)$) are generated in the example mentioned above. This is because activity v happens between activity b and c in some traces in $E^{'}$ infrequently and v belongs to a different activity cluster from b and c. As a result, three kinds of infrequent sub-trace $<a,b>$, $<v>$ and $<c,d>$ in $H-SSE^{'}$ are generated by our method. The Algorithm 2 will remove all infrequent high level activities and also the sub-logs related to these activities. A lot more activities like activity v might lead to the situation that a huge amount of process behaviors in the original event logs will get lost because of being distributed into many infrequent sub-logs in $H-SSE$ which then will be removed. In this paper we propose a trace merging approach (called Λ which appears in the step 5 of Algorithm 2 and helps preserve the process behaviors recorded in the original logs as many as possible) for fixing this problem by employing the following definitions:

Definition 7. *(merging mode)*
Let $SSE = \{E_{mc_1}, E_{mc_2}, \ldots, E_{mc_n}\}$ be a set of sub-logs output by stage 1 of Algorithm 1 executed on an event log E. Let st_1 and st_2 be two sub-traces from $E_{mc_k} \in SSE$, sa_1 be the starting activity of st_1 and ea_2 be the ending activity of st_2. The pair (st_1, st_2) is called a merging mode for E_{mc_k} if (1) $|st_1| < \beta \times |E_{mc_k}|$ and $|st_2| < \beta \times |E_{mc_k}|$ where $|st_1|$ represents the total number of traces in E_{mc_k} which have the same event sequence as st_1, $|st_2|$ represents the total number of traces which have the same event sequence as st_2 and $|E_{mc_k}|$ represents the total number of traces in E_{mc_k}, (2) st_1 and st_2 appear in the same trace from E in the way $<st_1, \ldots, st_2>$, (3) the number of traces in E_{mc_k} which have sa_1 as starting activity and ea_2 as ending activity at the same time is larger than or equal to $\beta \times |E_{mc_k}|$.

Definition 8. *(minimum merging mode)*
Let (st_1, st_2) be a merging mode for a sub-log $E_{mc_k} \in SSE$, sa_1 be the starting activity of st_1 and ea_2 be the ending activity of st_2, $<st_1, \ldots, st_2>$ be a sub-trace from the original log E. The merging mode (st_1, st_2) is called a minimum merging mode if there exists no other merging modes in the sub-trace $<st_1, \ldots |st_2>$ or in the sub-trace $<st_1 | \ldots, st_2>$, where $<st_1, \ldots |st_2>$ represents a sub-trace generated by removing st_2 from $<st_1, \ldots, st_2>$ and $<st_1 | \ldots, st_2>$ by removing st_1 from $<st_1, \ldots, st_2>$.

For the example mentioned above, given a threshold $\beta = 0.05$, the pair $(<a,b>, <c,d>)$ from $E^{'}_{mc_1}$ is a merging mode (there are eight of such merging modes) because there are 330 traces in $E^{'}_{mc_1}$ that have activity a as starting activity and activity d as ending activity which is larger than $\beta \times |E^{'}_{mc_1}| = 17.3$. In the meantime, $|<a,b>| = 8 < 17.3$ and $|<c,b>| = 8 < 17.3$. Furthermore, the way for the sub-traces $<a,b>$ and $<c,d>$ to appear in the trace $<a,b,v,c,d,u,x,z>$ from $E^{'}$ also satisfies the condition proposed in Definition 7. The merging mode $(<a,b>, <c,d>)$ is also a minimum merging mode according to Definition 8.

Algorithm 3: Merging Traces (Λ).

Input: the set of sub-logs SSE, a sub-log $E_{mc_k} \in SSE$, a threshold β.
Let SMD be a set of merging modes.
1: $SMD \leftarrow Null$
2: **for** each sub-trace $st_p \in E_{mc_k}$ **do**
3: **if** st_p doesn't pertain to any merging mode in SMD **then**
4: **if** there exists another sub-trace $st_q \in E_{mc_k}$ and (st_p, st_q) is a merging mode **then**
5: put (st_p, st_q) in SMD
6: put the related sub-trace $<st_p, \ldots, st_q>$ from E in E_{mc_k}
7: remove st_p and st_q from E_{mc_k}
8: remove the sub-traces that appear between st_p and st_q in $<st_p, \ldots, st_q>$ from their original places in SSE
9: **end if**
10: **else**
11: *continue*
12: **end if**
13: **end for**
Output: the sub-log E_{mc_k} with merged traces.

With the two definitions created above, the details of the trace merging technique Λ is described in Algorithm 3. Here we still use the last example to explain how Λ works. As is shown that three infrequent high level activities are generated by running the Algorithm 2 directly starting from step 7 in our example. One intuitive method to solve this problem is to find all minimum merging modes in $SSE^{'}$ and then merge the sub-traces in the same merging

mode (reflected by the steps $2-13$ of Algorithm 3 and the steps $4-9$ of Algorithm 2). For example, eight merging modes $(<a,b>,<c,d>)^8$ for E'_{mc_1} can be constituted (given a threshold $\beta = 0.05$) and each pair of the sub-traces should be merged into a single sub-trace $<a,b,v,c,d>$ (eight of such merged sub-traces can be generated). Then, a new set of sub-logs $M{-}SSE' = \{ME'_1, ME'_2\}$ can be formed, where $ME'_1 = \{<a,b,d>^{100}, <a,b,c,d>^{80}, <a,c,d>^{150}, <a,b,v,c,d>^8\}$ and $ME'_2 = \{<u,x,z>^{108}, <v,x,z>^{80}, <u,v,x,z>^{150}\}$ (ME'_2 doesn't contain the kind of sub-trace $<v>$ any more because all of them are merged into the kind of sub-trace $<a,b,v,c,d>$ in ME'_1). Afterwards, by using the steps $7-18$ of Algorithm 2 to deal with the $M{-}SSE'$ a new set of sub-logs $H{-}SSE' = \{\{<a,b,d>^{100}, <a,b,c,d>^{80}, <a,c,d>^{150}, <a,b,v,c,d>^8\}_0, \{<u,x,z>^{108}, <u,v,x,z>^{150}\}_1, \{<v,x,z>^{80}\}_2\}$ and a new set of high level activities $H{-}SA' = \{HL{-}Activity(0)^{338}, HL{-}Activity(1)^{258}, HL{-}Activity(2)^{80}\}$ can be generated. Now no infrequent high level activities exist in $H{-}SA'$ any longer.

3.2.3 Deal With Complex and Inaccurate Sub-models

In this subsection we presume that a set of sub-logs $H{-}SSE$ has been output by the stage 2 of Algorithm 1. For each sub-log in $H{-}SSE$ a sub-model is mined with existing workflow discovery technique to depict the details of the sub-log's relevant high level activity. In our approach, the business process recorded in an event log is expressed by the generated high level model and the sub-models together. However, the strategy (mentioned in Section 2) used in our method try to decrease the number of behaviors in the potential high level model by hiding most of the original process behaviors inside the high level activities generated. As a result, the sub-models for the high level activities might still be complex and inaccurate. Trace clustering technique is utilised for solving this problem.

Let $\Omega : S{-}E \rightarrow S{-}M$ be a workflow discovery algorithm, where $S{-}M$ is the set of process models and $S{-}E$ is the set of event logs, $\Theta_{accuracy} : (S{-}E, S{-}M) \rightarrow SV_{accuracy}$ be a process model accuracy evaluation method, where $SV_{accuracy}$ is the set of accuracy values of the mined process models, $\Theta_{complexity} : S{-}M \rightarrow SV_{complexity}$ be a process model complexity evaluation method, where $SV_{complexity}$ is the set of complexity values of the mined process models. Let $T_{clustering} : (S{-}E, SV_{cnumber}) \rightarrow SS{-}E$ be a trace clustering algorithm, $SS{-}E$ is the set of all sets of sub-logs and $SV_{cnumber}$ is the set of numbers of the clusters generated. The main procedure for dealing with the low-quality sub-models mined is depicted in Algorithm 4.

Algorithm 4: Deal with low-quality sub-models.

Input: the set of sub-logs $H{-}SSE$, a sub-model complexity threshold τ and a sub-model accuracy threshold χ, a trace number threshold κ, cluster number n.
Let SSM, SSM_c be two sets of sub-models.
Let $S{-}E_c$ be a set of sub-logs.
Let m_1, m_2 be two variants of float type.
Let m_3 be a variant of int type.
1: $SSM \leftarrow Null$, $SSM_c \leftarrow Null$
2: $S{-}E_c \leftarrow Null$
3: $m_1 \leftarrow 0$, $m_2 \leftarrow 0$
4: $m_3 \leftarrow 0$
5: **for** each sub-log $SE \in H{-}SSE$ **do**
6: **if** $\Theta_{accuracy}(\Omega(SE), SE) < \chi$ $||$ $\Theta_{complexity}(\Omega(SE), SE) > \tau$ && $|SE| \geq \kappa$ **then**
7: $S{-}E_c = T_{clustering}(SE, n)$
8: **for** each sub-log $SE_c \in S{-}E_c$ **do**
9: $SSM_c \leftarrow SSM_c \cup \Omega(SE_c)$
10: $m_1 \leftarrow m_1 + \Theta_{accuracy}(\Omega(SE_c), SE_c)|SE_c|$
11: $m_3 \leftarrow m_3 + |SE_c|$
12: **end for**
13: $m_2 \leftarrow m_1/m_3$ # calculate weighted average # accuracy
14: **if** $m_2 \geq \Theta_{accuracy}(\Omega(SE), SE)$ **then**
15: **for** each sub-model $SM_c \in SSM_c$ **do**
16: $SSM \leftarrow SSM \cup SM_c$
17: **end for**
18: **else**
19: $SSM \leftarrow SSM \cup \Omega(SE)$
20: **end if**
21: $m_1 \leftarrow 0$, $m_3 \leftarrow 0$, $SSM_c \leftarrow Null$
22: **else**
23: $SSM \leftarrow SSM \cup \Omega(SE)$
24: **end if**
25: **end for**
Output: the set of sub-models SSM.

According to Algorithm 4, a sub-log SE from $H{-}SSE$ that leads to a low-quality sub-model M (the quality is judged by using the sub-model accuracy and complexity thresholds χ and τ in the step 6 of Algorithm 4) will be divided into n sub-sub-logs by using the trace clustering technique if the number of the traces inside SE is larger than or equal to a threshold κ. Afterwards, for each sub-sub-log a sub-sub-model is built (in the step 9 of Algorithm 4). If the weighted average accuracy of the sub-sub-models generated is larger than or equal to the accuracy of the original

sub-model then these sub-sub-models are added to the set of sub-models *SSM* which will be finally output by Algorithm 4 (Algorithm 4 will not use the sub-sub-models if their weighed average accuracy is lower than the accuracy of their related original sub-model). If a sub-log $SE^{'}$ from $H{-}SSE$ leads to a good-quality sub-model $M^{'}$ then add $M^{'}$ in *SSM* (step 23 of Algorithm 4).

The authors in (Weerdt et al., 2013) develop a metric called *Place/Transition Connection Degree* (PT-CD) for quantifying the complexity of a Petri net which is defined as:

$$PT{-}CD = \frac{1}{2}\frac{|a|}{|P|} + \frac{1}{2}\frac{|a|}{|T|} \quad (2)$$

In Equation 2, $|a|$ represents the total number of arcs in the process model, $|P|$ is the number of places and $|T|$ is the number of transitions. The greater the PT-CD is, the more complicated the model will be.

In this paper, we utilise the *Heuristics Miner* (HM) (Weijters et al., 2006) for generating the process models. The ICS fitness developed in (de Medeiros, 2006) is utilised for evaluating the accuracy of the mined heuristic net. Then, the *Heuristic Net to Petri Net* plugin in ProM 6[2] is used for transforming the heuristic net mined into a Petri net. Afterwards, the *PT-CD* is employed for evaluating the complexity of the Petri net obtained. The trace clustering technique *GED* from (Bose and van der Aalst, 2009) is utilised for dividing the sub-logs in $H{-}SSE$ into sub-sub-logs (step 7 of Algorithm 4).

4 CASE STUDY

We tested the effectiveness of our approach on three event logs: the repair log (Repair) from (van der Aalst, 2011), the hospital log (Hospital) from BPIC 2011 (in our experiment an artificial start activity and end activity are added in the traces from the hospital log) and the log of the loan and overdraft approvals process (Loan) from BPIC 2012. The basic information about the three logs is shown in Table 1. The quality information of the models mined from the three logs by using HM is listed in Table 2. Except for *Place/Transition Connection Degree* (PT-CD) mentioned in the last section, another process model complexity metric is also used for evaluating the complexity of the mined models in our experiment which is *Extended Cardoso Metric* (E-Cardoso) (Lassen and van der Aalst, 2009).

Firstly, six classical trace clustering techniques are executed on the three logs which are 3-gram (Song et al., 2009), MR and MRA (Bose and van der Aalst, 2010), ATC (Weerdt et al., 2013), GED (Bose and van der Aalst, 2009) and sequence clustering (SC) (Ferreira et al., 2007). For each trace clustering approach six sub-logs are generated for every of the three logs utilised. The assessment results on these techniques are shown in Table 3. The metric $W_t - ICS$ stands for the weighted average ICS fitness based on the number of traces and $W_e - ICS$ represents the weighted average ICS fitness based on the number of events. For example, let $S{-}E = \{E_1, E_2, E_3, E_4, E_5, E_6\}$ be a set of sub-logs output by a trace clustering technique carried out on event log E. For a sub-log $E_k \in S{-}E$, $|E_k|_t$ represents the total number of traces in E_k, $|E_k|_e$ represents the total number of events in E_k and ICS_{E_k} represents the value of ICS fitness for the sub-model mined from sub-log E_k. Then, the $W_t - ICS$ for the sub-logs in $S{-}E$ is equal to $(\sum_{k=1}^{6} |E_k|_t \times ICS_{E_k}) / \sum_{k=1}^{6} |E_k|_t$ and the $W_e - ICS$ is equal to $(\sum_{k=1}^{6} |E_k|_e \times ICS_{E_k}) / \sum_{k=1}^{6} |E_k|_e$. According to the evaluation results shown in Table 3, most trace clustering techniques perform well on the log Repair which contains the least trace behaviors among the three logs. Nevertheless, for the logs Loan and Hospital which have more trace behaviors most trace clustering techniques employed could not bring a significant improvement on the accuracy of the mined models (especially for the log Hospital).

Table 1: Basic information of the evaluated logs.

Log	Traces	Events	Event types
Repair	1000	10827	12
Loan	13087	262200	36
Hospital	1143	150291	624

Table 2: Evaluation results for the models mined by using the log Repair, Loan and Hospital.

Log	ICS	$E-Cardoso$	$PT-CD$
Repair	0.6768	31	2.3656
Loan	0.7878	148	3.1478
Hospital	0.6058	2108	2.703

Whereafter, the approach proposed in this paper is evaluated by using the three logs mentioned above. The threshold ϕ for judging the casual relations is set to zero (such a setting will help find more complete activity clusters), the threshold α for judging whether a high level activity generated should be removed or not is set to 20, the threshold β for searching for the merging modes is set to 0.05, the sub-model complexity threshold τ (for PT-CD) is set to 2.5, the sub-model accuracy threshold χ (for ICS fitness) is set to 0.8, the trace number threshold κ is set to 100 and the number of clusters for the trace clustering technique *GED* is

[2]http://www.promtools.org.

Table 3: Evaluation results for the six classical trace clustering techniques executed on the log Repair, Loan and Hospital.

Log	Method	$W_t - ICS$	$W_e - ICS$
Repair	3-gram	0.9299	0.9326
	MR	0.8123	0.814
	MRA	0.8056	0.8055
	ATC	0.9971	0.996
	GED	0.7908	0.7907
	SC	0.9823	0.9802
Loan	3-gram	0.7965	0.7282
	MR	0.7828	0.6984
	MRA	0.8181	0.7285
	ATC	0.7653	0.5665
	GED	0.8038	0.7992
	SC	0.9255	0.9164
Hospital	3-gram	0.6153	0.69
	MR	0.5785	0.6622
	MRA	0.5629	0.6844
	ATC	0.7583	0.705
	GED	0.6003	0.6837
	SC	0.7354	0.7129

set to 6. The quality information of the sub-models generated is shown in Table 4, the quality information of the three high level models (for the log Repair, Loan and Hospital) output by our technique is shown in Table 5 and the basic information of the three high level logs created by our technique is shown in Table 6.

According to Table 6, the generated high level logs H-Repair and H-Hospital contains fewer activities than their related raw event logs Repair and Hospital. The main reason is that the activities in the original repair log and hospital log can form high quality activity clusters (more activity relations inside the cluster and fewer among the clusters). In the experiment about 1% events from log Hospital and 0.5% events from log Loan are removed together with the infrequent high level activities generated and for the log Repair no events are removed (very few events are removed because of the effects of the trace merging technique proposed in Section 3).

According to Table 5, all of the three high level models generated have high accuracy which benefits from the abstraction strategy put forward in Section 2. For the high level activities in the three built high level models, the average accuracy of their relevant sub-models is also generally good.

5 RELATED WORK

Trace clustering technique is one of the most effective approaches for dealing with the negative impacts from high variety of behaviors recorded in event logs. Several classical trace clustering approaches have been proposed in the literature. In (Song et al., 2009) the authors put forward an approach which is able to abstract the features of the traces from event logs into five profiles that includes *activity profile*, *transition profile*, *case attributes profile*, *event attributes profile* and *performance profile*. Afterwards, these profiles are converted into an aggregate vector so that the distance between any two traces can be measured. The main advantage of this technique is that it considers a complete range of metrics for clustering traces. In (Bose and van der Aalst, 2010) and (Bose and van der Aalst, 2009) the context-aware trace clustering techniques are proposed which try to improve the output results of trace clustering by employing the context knowledge that can be acquired from event logs. In (Bose and van der Aalst, 2010) the authors point out that the feature sets based on sub-sequences of traces are context-aware and can express some process functions. The traces that have many common conserved features should be put in the same cluster. The authors in (Bose and van der Aalst, 2009) develop an edit distance-based trace clustering algorithm. The context knowledge mined from event logs are integrated in the calculation procedure for the cost of edit operations. The Markov trace clustering method is put forward in (Ferreira et al., 2007). This method calculates a potential first-order Markov model for each cluster based on an expectation-maximization algorithm. A trace is sent to a cluster which has a Markov model that can generate this trace with a high probability. In (Weerdt et al., 2013) a novel technique named *active trace clustering* is presented. This technique tries to optimise the fitness of each cluster's underlying process model during the run time without employing the vector space model for the clustering process. It simply distributes the traces to the suitable clusters by considering the optimization of the combined accuracy of the potential models for these clusters. Most trace clustering techniques perform well for dealing with the event logs with a moderate amount of trace behaviors. However, such techniques can not assure a good result while being executed on the logs with massive behaviors (as is shown in the case study in Section 4).

Process model abstraction approach is also effective for dealing with the "spaghetti-like" business process models mined. In (Bose and van der Aalst, 2009) the authors develop a two-step approach for mining

Table 4: The weighted average quality of the sub-models generated by our method.

Log	$W_t - ICS$	$W_e - ICS$	$W_t - E{-}Cardoso$	$W_e - E{-}Cardoso$	$W_t - PT{-}CD$	$W_e - PT{-}CD$
Repair	0.9738	0.9687	11.57	12.46	2.0688	2.0929
Loan	0.9514	0.9297	21.934	26.4995	2.1729	2.2238
Hospital	0.8891	0.902	467.84	465.2	3.1257	3.0956

Table 5: The quality information of the high level models generated for each log by our technique.

Log	ICS	$E - Cardoso$	$PT{-}CD$
Repair	0.978	33	2.483
Loan	0.9671	137	3.378
Hospital	0.95	192	2.4328

Table 6: Basic information of the generated high level logs.

H-Log	**Traces**	**Events**	**Event types**
H-Repair	1000	2700	10
H-Loan	13087	40783	44
H-Hospital	1143	37740	65

hierarchical business process models. This approach searches for the sub-traces that repeatedly happen in event logs. Two kinds of such sub-traces are defined which are *tandem arrays* and *maximal repeats*. This approach firstly searches for all the *tandem arrays* and the *maximal repeats* in the event logs and then replace them in the original event logs by using high level activities (each high level activity is an abstraction of a *tandem array* or a *maximal repeat* found) so that the high level event logs can be generated. Finally, the high level models (more accurate and simpler) could be mined by using existing workflow discovery algorithms executed on the high level logs. The authors in (Baier and Mendling, 2013) indicate that the low level events recorded in the event logs may be too granular and should be mapped to the high level activities predefined in the enterprise process specifications. Hence, they put forward a mapping method that combines the domain knowledge captured from these specifications. With the high level activities generated the better models on the higher abstraction level can be built. The authors in (Conforti et al., 2014) present an automated technique for mining the BPMN models with subprocesses. This technique analyses the dependencies among the data attributes attached to events. The events that are judged to have high dependencies will be put in the same subprocesses. Most of the classical process model abstraction approaches presented focus mainly on searching for the subprocesses and can not assure the quality of the built high level models. It is possible that the high level activities in the underlying abstracted models may still have a large amount of relations among each other.

6 CONCLUSION

In this paper we proposed a new method which combines the characters of the classical model abstraction techniques and the trace clustering techniques for solving the problem of inaccurate and complex process models mined. This method is able to optimise the quality of the underlying high level models through an efficient abstraction strategy and also considers the quality of the sub-models generated through trace clustering techniques. Finally, the details of the business processes recorded in the event logs are revealed by the high level models built together with the generated sub-models where each sub-model shows the details of its relevant high level activity. Though the results of the case study we demonstrated the effectiveness of our technique.

Our future work will mainly be focused on developing new trace clustering techniques with higher performance to help deal with the complex and inaccurate sub-models generated for the high level activities. We will also validate our method on some other real-life cases.

REFERENCES

van der Aalst, W., van Dongen, B. F., Herbst, J., Maruster, L., Schimm, G. and Weijters, A. J. M. M. (2003). Workflow Mining: A Survey of Issues and Aproaches. In *Data and Knowledge Engineering*, 47(2): 237–267, 2003.

van der Aalst, W. (2011). *Process Mining: Discovery, Conformance and Enhancement of Business Processes*. Springer-Verlag Berlin Heidelberg, 1nd edition.

Weerdt, J. D., vanden Broucke, S., Vanthienen, J., and Baesens, B. (2013). Active Trace Clustering for Improved Process Discovery. *IEEE Transactions on Knowledge and Data Engineering*, 25(12):2708–2720.

Bose, R. and van der Aalst, W. (2009). Context Aware Trace Clustering: Towards Improving Process Mining Results. In *Proceedings of the SIAM International Conference on Data Mining*, pages 401–412.

Bose, R. and van der Aalst, W. (2010). Trace Clustering Based on Conserved Patterns: Towards Achieving Better Process Models. In *Business Process Management Workshops*, volume 43 of *Lecture Notes in Business Information Processing*, pages 170–181. Springer Berlin.

Song, M., Gnther, C. and van der Aalst, W. (2009). Trace Clustering in Process Mining. In *Business Process Management Workshops*, volume 17 of *Lecture Notes in Business Information Processing*, pages 109–120. Springer Berlin.

Ferreira, D. R., Zacarias, M., Malheiros, M.and Ferreira, P. (2007). Approaching Process Mining with Sequence Clustering: Experiments and Findings. In *Business Process Management (BPM 2007)*, volume 4714 of *Lecture Notes in Computer Science*, pages 360–374.

Bose, R. and van der Aalst, W. (2009). Abstractions in Process Mining: A Taxonomy of Patterns. In *Business Process Management (BPM 2009)*, volume 5701 of *Lecture Notes in Computer Science*, pages 159–175.

Baier, T. and Mendling, J. (2013). Bridging Abstraction Layers in Process Mining: Event to Activity Mapping. In *BPMDS 2013 and EMMSAD 2013*, volume 147 of *Lecture Notes in Business Information Processing*, pages 109–123.

Conforti, R., Dumas, M., Carcia-Banuelos, L. and Rosa, M. L. (2014). Beyond Tasks and Gateways: Discovering BPMN Models with Subprocesses, Boundary Events and Activity Markers. In *Business Process Management (BPM 2014)*, volume 8659 of *Lecture Notes in Computer Science*, pages 101–117.

Schaeffer, S. E. (2007). Graph Clustering. In *Computer Science Review*, 1(1):27–64, 2007.

Hompes, B. F. A., Verbeek, H. M. W. and van der Aalst, W. (2014). Finding Suitable Activity Clusters for Decomposed Process Discovery. In *Proc. of the 4th Int'l. Symp. on Data-driven Process Discovery and Analysis*, volume 1293 of *CEUR Workshop Proceeding*, pages 16–30.

Weijters, A., van der Aalst, W. and Alves de Medeiros, A. K. (2006). Process Mining with the Heuristics Algorithm. In *BETA Working Paper Series*, 166, 2006.

Noack, A. (2007). Energy Models for Graph Clustering. In *J. Graph Algorithms Appl*, 11(2):453–480, 2007.

de Medeiros, A. A. (2006). Genetic Process Mining. In *PhD. thesis, Eindhoven University of Technology*, 2006.

Lassen, K. B. and van der Aalst, W. (2009). Complexity Metrics for Workflow Nets. In *Inform. Software Technol.*, 51:610–626.

SJClust: Towards a Framework for Integrating Similarity Join Algorithms and Clustering

Leonardo Andrade Ribeiro[1], Alfredo Cuzzocrea[2],
Karen Aline Alves Bezerra[3] and Ben Hur Bahia do Nascimento[3]
[1]*Instituto de Informática, Universidade Federal de Goiás, Goiânia, Brazil*
[2]*University of Trieste and ICAR-CNR, Trieste, Italy*
[3]*Departamento de Ciência da Computação, Universidade Federal de Lavras, Lavras, Brazil*
laribeiro@inf.ufg.br, alfredo.cuzzocrea@dia.units.it, karen.bezerra@posgrad.ufla.br, bhn@computacao.ufla.br

Keywords: Data Integration, Data Cleaning, Duplicate Identification, Set Similarity Joins, Clustering.

Abstract: A critical task in data cleaning and integration is the identification of duplicate records representing the same real-world entity. A popular approach to duplicate identification employs similarity join to find pairs of similar records followed by a clustering algorithm to group together records that refer to the same entity. However, the clustering algorithm is strictly used as a post-processing step, which slows down the overall performance and only produces results at the end of the whole process. In this paper, we propose SjClust, a framework to integrate similarity join and clustering into a single operation. Our approach allows to smoothly accommodating a variety of cluster representation and merging strategies into set similarity join algorithms, while fully leveraging state-of-the-art optimization techniques.

1 INTRODUCTION

The presence of multiple records representing the same real-world entity plagues practically every large database. Such records are often referred to as *fuzzy duplicates* (duplicates, for short), because they might not be exact copies of one another. Duplicates arise due to a variety of reasons, such as typographical errors and misspellings during data entry, different naming conventions, and as a result of the integration of data sources storing overlapping information.

Duplicates degrade the quality of the data delivered to application programs, thereby leading to a myriad of problems. Some examples are misleading data mining models owing to erroneously inflated statistics, inability of correlating information related to a same entity, and unnecessarily repeated operations, e.g., mailing, billing, and leasing of equipment. Duplicate identification is thus of crucial importance in data cleaning and integration.

Duplicate identification is computationally very expensive and, therefore, typically done offline. However, there exist important application scenarios that demand (near) real-time identification of duplicates. Prominent examples are data exploration (Idreos et al., 2015), where new knowledge has to be efficiently extracted from databases without a clear definition of the information need, and virtual data integration (Doan et al., 2012), where the integrated data is not materialized and duplicates in the query result assembled from multiple data sources have to be identified — and eliminated — on-the-fly. Such scenarios have fueled the desire to integrate duplicate identification with processing of complex queries (Altwaijry et al., 2015) or even as a general-purpose physical operator within a DBMS (Chaudhuri et al., 2006).

An approach to realize the above endeavor is to employ *similarity join* in concert with a *clustering algorithm* (Hassanzadeh et al., 2009). Specifically, similarity join is used to find all pairs of records whose similarity is not less than a specified threshold; the similarity between two records is determined by a *similarity function*. In a post-processing step, the clustering algorithm groups together records using the similarity join results as input.

For data of string type, *set similarity join* is an appealing choice for composing a duplicate identification operator. Set similarity join views its operands as sets — strings can be easily mapped to sets. The corresponding similarity function assesses the similarity between two sets in terms of their overlap and a rich variety of similarity notions can be expressed in this way (Chaudhuri et al., 2006). Furthermore, a number of optimization techniques have been proposed

Ribeiro, L., Cuzzocrea, A., Bezerra, K. and Nascimento, B.
SJClust: Towards a Framework for Integrating Similarity Join Algorithms and Clustering.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 75-80
ISBN: 978-989-758-187-8

over the years (Sarawagi and Kirpal, 2004; Chaudhuri et al., 2006; Bayardo et al., 2007; Xiao et al., 2008; Ribeiro and Härder, 2011) yielding highly efficient and scalable algorithms.

The strategy of using a clustering algorithm strictly for post-processing the results of set similarity join has two serious drawbacks, however. First, given a group of n, sufficiently similar, duplicates, the set similarity join performs $\binom{n}{2}$ similarity calculations to return the same number of set pairs. While this is the expected behavior considering a similarity join in isolation, it also means that repeated computations are being performed over identical subsets. Even worse, we may have to perform much more additional similarity calculations between non-duplicates: low threshold values are typically required for clustering algorithms to produce accurate results (Hassanzadeh et al., 2009). Unfortunately, existing filtering techniques are not effective at low threshold values and, thus, there is an explosion of the number of the comparison at such values. Second, the clustering is a blocking operator in our context, i.e., it has to consume all the similarity join output before producing any cluster of duplicates as result element. This fact is particularly undesirable when duplicate identification is part of more complex data processing logic, possibly even with human interaction, because it prevents pipelined execution.

In this paper, we present *SJClust*, a framework to integrate set similarity join and clustering into a single operation, which addresses the above issues. The main idea behind our framework is to represent groups of similar sets by a *cluster representative*, which is incrementally updated during the set similarity join processing. Besides effectively reducing the number similarity calculations needed to produce a cluster of n sets to $O(n)$, we are able to fully leverage state-of-the-art optimization techniques at high threshold values, while still performing well at low threshold values where such techniques are less effective. Moreover, we exploit set size information to identify when no new set can be added to a cluster; therefore, we can then immediately output this cluster and, thus, avoid the blocking behavior. On the other hand, improving performance of clustering algorithms is critical for next-generation big data management and analytics applications (e.g., (Cuzzocrea et al., 2013b; Cuzzocrea, 2013; Cuzzocrea et al., 2013a)).

Furthermore, there exists a plethora of clustering algorithms suitable for duplicate identification and no single algorithm is overall the best across all scenarios (Hassanzadeh et al., 2009). Thus, versatility in supporting a variety of clustering methods is essential. Our framework smoothly accommodates various cluster representation and merging strategies, thereby yielding different clustering methods for each combination thereof.

2 BASIC CONCEPTS AND DEFINITIONS

In this section, we present important concepts and definitions related to set similarity joins before present important optmization techniques.

We map strings to *sets of tokens* using the popular concept of *q-grams*, i.e., sub-strings of length q obtained by "sliding" a window over the characters of an input string v. We (conceptually) extend v by prefixing and suffixing it with $q-1$ occurrences of a special character *"$"* not appearing in any string. Thus, all characters of v participate in exact q q-grams. For example, the string *"token"* can be mapped to the set of *2*-gram tokens $\{$*t*, *to*, *ok*, *ke*, *en*, *n*$\}$. As the result can be a multi-set, we simply append the symbol of a sequential ordinal number to each occurrence of a token to convert multi-sets into sets, e.g, the multi-set $\{$a,b,b$\}$ is converted to $\{a\circ 1, b\circ 1, b\circ 2\}$. In the following, we assume that all strings in the database have already been mapped to sets.

We associate a weight with each token to obtain *weighted sets*. A widely adopted weighting scheme is the Inverse Document Frequency (*IDF*), which associates a weight $idf(tk)$ to a token tk as follows: $idf(tk)$=$ln(1+N/df(tk))$, where $df(tk)$ is the *document frequency*, i.e., the number of strings a token tk appears in a database of N strings. The intuition behind using IDF is that rare tokens are more discriminative and thus more important for similarity assessment. We obtain *unweighted sets* by associating the value 1 to each token. The weight of a set r, denoted by $w(r)$, is given by the weight summation of its tokens, i.e., $w(r)=\sum_{tk\in r} w(tk)$; note that we have $w(r)=|r|$ for unweighted sets.

We consider the general class of set similarity functions. Given two sets r and s, a set similarity function $sim(r,s)$ returns a value in $[0,1]$ to represent their similarity; larger value indicates that r and s have higher similarity. Popular set similarity functions are defined as follows.

Definition 1 (Set Similarity Functions)**.** *Let r and s be two sets. We have:*

- *Jaccard similarity:* $J(r,s)=\frac{w(r\cap s)}{w(r\cup s)}$.
- *Dice similarity:* $D(r,s)=\frac{2\cdot w(r\cap s)}{w(r)+w(s)}$.
- *Cosine similarity:* $C(r,s)=\frac{w(r\cap s)}{\sqrt{w(r)\cdot w(s)}}$

We now formally define the set similarity join operation.

Definition 2 (Set Similarity Join)**.** *Given two set collections $\mathcal{R}$ and $\mathcal{S}$, a set similarity function sim, and a threshold τ, the set similarity join between $\mathcal{R}$ and $\mathcal{S}$ returns all scored set pairs $\langle (r,s), \tau\prime \rangle$ s.t. $(r,s) \in \mathcal{R} \times \mathcal{S}$ and $sim(r,s) = \tau\prime \geq \tau$.*

In this paper, we focus on self-join, i.e., $\mathcal{R} = \mathcal{S}$; we discuss the extension for binary inputs in Section 4. For brevity, we use henceforth the term similarity function (join) to mean set similarity function (join). Further, we focus on the Jaccard similarity and the IDF weighting scheme, i.e., unless stated otherwise, $sim(r,s)$ and $w(tk)$ denotes $J(r,s)$ and $idf(tk)$, respectively.

Example 1. *Consider the sets r and s below*

$$x = \{\boldsymbol{A}, \boldsymbol{B}, C, \boldsymbol{D}, \boldsymbol{E}\}$$
$$y = \{\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{D}, \boldsymbol{E}, F\}$$

and the following token-IDF association table:

tk	A	B	C	D	E	F
$idf(tk)$	1.5	2.5	2	3.5	0.5	2

We have $w(r) = w(s) = 10$ and $w(r \cap s) = 8$; thus $sim(r,s) = \frac{8}{10+10-8} \approx 0.66$.

3 OPTIMIZATION TECHNIQUES

In this section, we describe a general set similarity join algorithm, which provides the basis for our framework.

Similarity functions can be equivalently represented in terms of an *overlap bound* (Chaudhuri et al., 2006). Formally, the overlap bound between two sets r and s, denoted by $O(r,s)$, is a function that maps a threshold τ and the set weights to a real value, s.t. $sim(r,s) \geq \tau$ iff $w(r \cap s) \geq O(r,s)$[1]. The similarity join can then be reduced to the problem of identifying all pairs r and s whose overlap is not less than $O(r,s)$. For the Jaccard similarity, we have $O(r,s) = \frac{\tau}{1+\tau} \cdot (w(r) + w(s))$.

Further, similar sets have, in general, roughly similar weights. We can derive bounds for immediate pruning of candidate pairs whose weights differ enough. Formally, the weight bounds of r, denoted by $min(r)$ and $max(r)$, are functions that map τ and $w(r)$ to a real value s.t. $\forall s$, if $sim(r,s) \geq \tau$, then $min(r) \leq w(s) \leq max(r)$ (Sarawagi and Kirpal, 2004). Thus, given a set r, we can safely ignore all other sets whose weights do not fall within the interval $[min(r), max(r)]$. For the Jaccard similarity, we $[min(r), max(r)] = \left[\tau \cdot w(r), \frac{w(r)}{\tau}\right]$. We refer the reader to (Schneider et al., 2015) for definitions of overlap and weight bounds of several other similarity functions, including Dice and Cosine.

We can prune a large share of the comparison space by exploiting the *prefix filtering principle* (Sarawagi and Kirpal, 2004; Chaudhuri et al., 2006). Prefixes allow selecting or discarding candidate pairs by examining only a fraction of the original sets. We first fix a global order O on the universe $\mathcal{U}$ from which all tokens in the sets considered are drawn. A set $r' \subseteq r$ is a prefix of r if r' contains the first $|r'|$ tokens of r. Further, $pref_\beta(r)$ is the shortest prefix of r, the weights of whose tokens add up to more than β. The prefix filtering principle is defined as follows.

Definition 3 (Prefix Filtering Principle (Chaudhuri et al., 2006))**.** *Let r and s be two sets. If $w(r \cap s) \geq \alpha$, then $pref_{\beta_r}(r) \cap pref_{\beta_s}(r) \neq \varnothing$, where $\beta_r = w(r) - \alpha$ and $\beta_s = w(s) - \alpha$, respectively.*

We can identify all candidate matches of a given set r using the prefix $pref_\beta(r)$, where $\beta = w(r) - min(r)$. We denote this prefix simply by $pref(r)$. It is possible to derive smaller prefixes for r, and thus obtain more pruning power, when we have information about the set weight of the candidate sets, i.e., if $w(s) \geq w(r)$ (Bayardo et al., 2007) or $w(s) > w(r)$ (Ribeiro and Härder, 2011). Note that prefix overlap is a condition necessary, but not sufficient to satisfy the original overlap constraint: an additional verification must be performed on the candidate pairs.

Further, the number of candidates can be significantly reduced by using the *inverse document frequency ordering*, O_{idf}, as global token order to obtain sets ordered by decreasing IDF weight [2]. The idea is to minimize the number of sets agreeing on prefix elements and, in turn, candidate pairs by shifting lower frequency tokens to the prefix positions.

Example 2. *Consider the sets r and s in Example 1 and $\tau = 0.6$. We have $O(r,s) = 7.5$; $[min(r), max(r)]$ and $[min(s), max(s)]$ are both $[6, 16.7]$. By ordering r and s according to O_{idf} and the IDF weights in Example 1, we obtain:*

$$x = [D, B, C, A, E]$$
$$y = [D, B, F, A, E].$$

We have $pref(r) = pref(s) = [D]$.

[1] For ease of notation, the parameter τ is omitted.

[2] A secondary ordering is used to break ties consistently (e.g., the lexicographic ordering). Also, note that an equivalent ordering is the *document frequency ordering*, which can be used to obtain unweighted sets ordered by increasing token frequency in the collection.

4 THE SIMILARITY JOIN ALGORITHM

In this section, we provide the details on the similarity join algorithm.

Similarity join algorithms based on inverted lists are effective in exploiting the previous optimizations (Sarawagi and Kirpal, 2004; Bayardo et al., 2007; Xiao et al., 2008; Ribeiro and Härder, 2011). Most of such algorithms have a common high-level structure following a filter-and-refine approach.

Algorithm 1 formalizes the steps of a similarity join algorith. The algorithm receives as input a set collection sorted in increasing order of set weights, where each set is sorted according to O_{idf}. An inverted list I_t stores all sets containing a token t in their prefix. The input collection R is scanned and, for each *probe set* r, its prefix tokens are used to find *candidate sets* in the corresponding inverted lists (lines 4–10); this is the *candidate generation phase*, where the map M is used to associate candidates to its accumulated overlap score *os* (line 3). Each candidate s is dynamically removed from the inverted list if its weight is less than $min(r)$ (lines 6–7). Further filters, e.g., filter based on overlap bound, are used to check whether s can be a true match for r, and then the overlap score is accumulated, or not, and s can be safely ignored in the following processing (lines 8–10). In the *verification phase*, r and its matching candidates are checked against the similarity predicate and those pairs satisfying the predicate are added to the result set. To this end, the *Verify* procedure (not shown) employs a merge-join-based algorithm exploiting token order and the overlap bound to define break conditions (line 11). Finally, in the *indexing phase*, a *pointer* to set r is appended to each inverted list I_t associated with its prefix tokens (lines 12 and 13).

Algorithm 1 is actually a self-join. Its extension to binary joins is trivial: we first index the smaller collection and then go through the larger collection to identify matching pairs. For simplicity, several filtering strategies such positional filtering (Xiao et al., 2008) and min-prefixes (Ribeiro and Härder, 2011), as well as inverted list reduction techniques (Bayardo et al., 2007; Ribeiro and Härder, 2011) were omitted. Nevertheless, these optimizations are based on bounds and prefixes and, therefore, our discussion in the following remains valid.

Algorithm 1: Similarity join algorithm.

Input: A set collection $\mathcal{R}$ sorted in increasing order of the set weight; each set is sorted according to O_{idf}; a threshold τ
Output: A set S containing all pairs (r,s) s.t. $Sim(r,s) \geq \tau$

```
1  I_1, I_2, ... I_|U| ← ∅, S ← ∅
2  foreach r ∈ R do
3      M ← empty map from set id to overlap score (os)
4      foreach t ∈ pref(r) do // can. gen. phase
5          foreach s ∈ I_t do
6              if w(s) < min(r)
7                  Remove s from I_t
8              if filter(r,s,M(s))
9                  M(s).os ← −∞        // invalidate s
10             else M(s).os = M(s).os + w(t)
11     S ← S ∪ Verify(r,M,τ)           // verif. phase
12     foreach t ∈ pref(r) do          // index. phase
13         I_t ← I_t ∪ {r}
14 return S
```

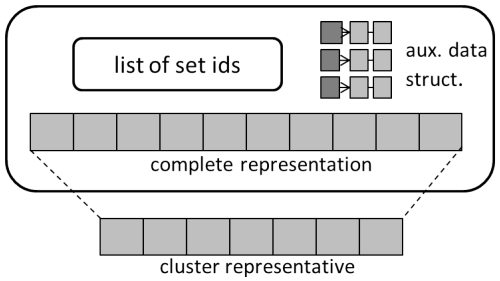


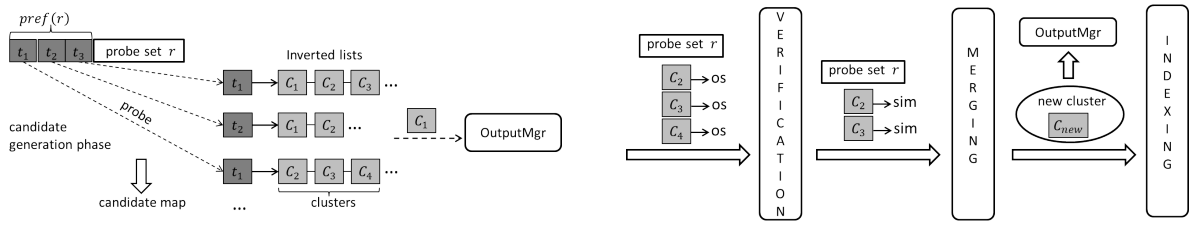
Figure 1: Cluster representation.

5 THE SJClust FRAMEWORK

We now present *SJClust*, a general framework to integrate clustering methods into similarity joins algorithms. The goals of our framework are threefold: 1) *flexibility and extensibility* to accommodate different clustering methods; 2) *efficiency* by fully leveraging existing optimization techniques and by reducing the number of similarity computations to form clusters; 3) *non-blocking behavior* by producing results before having consumed all the input.

The backbone of SJClust is the similarity join algorithm presented in the previous section. In particular, SJClust operates over the same input of sorted sets, without requiring any pre-processing, and has the three execution phases present in Algorithm 1, namely, candidate generation, verification, and indexing phases. Nevertheless, there are, of course, major differences.

First and foremost, the main objects are now cluster of sets, or simply clusters. Figure 1 illustrates strategy adopted for cluster representation. The internal representation contains a list of its set element's ids, an (optional) auxiliary structure, and the cluster's *complete representation*, a set containing all tokens from all set elements. The cluster export its external



(a) Candidate generation.

(b) SJClust course of evaluation.

Figure 2: SJClust framemork components.

representation as the so-called *cluster representative* (or simply representative), which is fully comparable to input sets. Similarity evaluations are always performed on the representatives, either between a probe set and a cluster or between two clusters. In the following, we use the term cluster and representative interchangeably whenever the distinction is unimportant for the discussion.

Figure 2 depicts more details on the SJClust framework. In the candidate generation phase, prefix tokens of the current probe set are used to find cluster candidates in the inverted lists (Figure 2(a)). Also, there is a *merging phase* between verification and indexing phases (Figure 2(b)). The verification phase reduces the number of candidates by removing false positives, i.e., clusters whose similarity to the probe set is less than the specified threshold. In the merging phase, a new cluster is generated from the probing set and the clusters that passed through the verification are considered for merging with it according to a *merging strategy*. In the indexing phase, references to the newly generated cluster are stored in the inverted lists associated with its prefix tokens. Finally, there is the so-called *Output Manager*, which is responsible for maintaining references to all clusters —a reference to a cluster is added to the Output Manager right after its generation in the merging phase (Figure 2(b)). Further, the Output Manager sends a cluster to the output as soon as it is identified that no new probing set can be similar to this cluster. Clusters in such situation can be found in the inverted lists during the candidate generation (Figure 2(a)) as well as identified using the weight of the probe set (not shown in Figure 2).

The aforementioned goals of SJClust are met as follows: flexibility and extensibility are provided by different combinations of cluster representation and merging strategies, which can be independently and transparently plugged into the main algorithm; efficiency is obtained by the general strategy to cluster representation and indexing; and non-blocking behavior is ensured by the Output Manager.

6 RELATED WORK

The duplicate identification problem has a long history of investigation conducted by several research communities spanning databases, machine learning, and statistics, frequently under different names, including record linkage, deduplication, and near-duplicate identification (Koudas et al., 2006; Elmagarmid et al., 2007). Over the last years, there is growing interest in realizing duplicate identification on-the-fly. In (Altwaijry et al., 2013), a query-driven approach is proposed to reduce the number of cleaning steps in simple selections queries over dirty data. The same authors presented a framework to answer complex Select-Project-Join queries (Altwaijry et al., 2015). Our work is complementary to these proposals as our framework can be encapsulated into physical operators to compose query evaluation plans.

There is long line of research on (exact) set similarity joins (Sarawagi and Kirpal, 2004; Chaudhuri et al., 2006; Bayardo et al., 2007; Xiao et al., 2008; Ribeiro and Härder, 2009; Ribeiro and Härder, 2011; Wang et al., 2012). Aspects most relevant to our work have already been discussed at length in Section 2. To the best of our knowledge, integration of clustering into set similarity joins has not been investigated in previous work.

In (Mazeika and Böhlen, 2006), the authors employ the concept of proximity graph to cluster strings without requiring a predefined threshold value. The algorithm to automatically detected cluster borders was improved later in (Kazimianec and Augsten, 2011). However, it is not clear how to leverage state-of-the-art set similarity joins in these approaches to improve efficiency and deal with large datasets.

In (Hassanzadeh et al., 2009), a large number of clustering algorithms are evaluated in the context of duplicate identification. These algorithms use similarity join to produce their input, but can start only after the execution of the similarity join.

7 CONCLUSIONS AND FUTURE WORK

In this paper, we presented SJClust, a framework to integrate clustering into set similarity join algorithms. Our framework provides flexibility and extensibility to accommodate different clustering methods, while fully leveraging existing optimization techniques and avoiding undesirable blocking behavior.

Future work is mainly oriented towards enriching our framework with advanced features such as uncertain data management (e.g., (Leung et al., 2013)), adaptiveness (e.g., (Cannataro et al., 2002)), and execution time prediction (e.g, (Sidney et al., 2015)).

ACKNOWLEDGEMENTS

This research was partially supported by the Brazilian agencies CNPq and CAPES.

REFERENCES

Altwaijry, H., Kalashnikov, D. V., and Mehrotra, S. (2013). Query-driven approach to entity resolution. *PVLDB*, 6(14):1846–1857.

Altwaijry, H., Mehrotra, S., and Kalashnikov, D. V. (2015). Query: A framework for integrating entity resolution with query processing. *PVLDB*, 9(3):120–131.

Bayardo, R. J., Ma, Y., and Srikant, R. (2007). Scaling up all pairs similarity search. In *Proc. of the 16th Intl. Conf. on World Wide Web*, pages 131–140.

Cannataro, M., Cuzzocrea, A., Mastroianni, C., Ortale, R., and Pugliese, A. (2002). Modeling adaptive hypermedia with an object-oriented approach and xml. *Second International Workshop on Web Dynamics*.

Chaudhuri, S., Ganti, V., and Kaushik, R. (2006). A primitive operator for similarity joins in data cleaning. In *Proc. of the 22nd Intl. Conf. on Data Engineering*, page 5.

Cuzzocrea, A. (2013). Analytics over big data: Exploring the convergence of datawarehousing, OLAP and data-intensive cloud infrastructures. In *37th Annual IEEE Computer Software and Applications Conference, COMPSAC 2013, Kyoto, Japan, July 22-26, 2013*, pages 481–483.

Cuzzocrea, A., Bellatreche, L., and Song, I. (2013a). Data warehousing and OLAP over big data: current challenges and future research directions. In *Proceedings of the sixteenth international workshop on Data warehousing and OLAP, DOLAP 2013, San Francisco, CA, USA, October 28, 2013*, pages 67–70.

Cuzzocrea, A., Saccà, D., and Ullman, J. D. (2013b). Big data: a research agenda. In *17th International Database Engineering & Applications Symposium, IDEAS '13, Barcelona, Spain - October 09 - 11, 2013*, pages 198–203.

Doan, A., Halevy, A. Y., and Ives, Z. G. (2012). *Principles of Data Integration*. Morgan Kaufmann.

Elmagarmid, A. K., Ipeirotis, P. G., and Verykios, V. S. (2007). Duplicate record detection: A survey. *TKDE*, 19(1):1–16.

Hassanzadeh, O., Chiang, F., Miller, R. J., and Lee, H. C. (2009). Framework for evaluating clustering algorithms in duplicate detection. *PVLDB*, 2(1):1282–1293.

Idreos, S., Papaemmanouil, O., and Chaudhuri, S. (2015). Overview of data exploration techniques. In *Proc. of the SIGMOD Conference*, pages 277–281.

Kazimianec, M. and Augsten, N. (2011). Pg-skip: Proximity graph based clustering of long strings. In *Proc. of the DASFAA Conference*, pages 31–46.

Koudas, N., Sarawagi, S., and Srivastava, D. (2006). Record linkage: Similarity measures and algorithms. In *Proc. of the SIGMOD Conference*, pages 802–803.

Leung, C. K., Cuzzocrea, A., and Jiang, F. (2013). Discovering frequent patterns from uncertain data streams with time-fading and landmark models. *T. Large-Scale Data- and Knowledge-Centered Systems*, 8:174–196.

Mazeika, A. and Böhlen, M. H. (2006). Cleansing databases of misspelled proper nouns. In *Proc. of the First Int'l VLDB Workshop on Clean Databases*.

Ribeiro, L. and Härder, T. (2009). Efficient set similarity joins using min-prefixes. In *Proc. of ADBIS Conference*, pages 88–102.

Ribeiro, L. A. and Härder, T. (2011). Generalizing prefix filtering to improve set similarity joins. *Information Systems*, 36(1):62–78.

Sarawagi, S. and Kirpal, A. (2004). Efficient set joins on similarity predicates. In *Proc. of the SIGMOD Conference*, pages 743–754.

Schneider, N. C., Ribeiro, L. A., de Souza Inácio, A., Wagner, H. M., and von Wangenheim, A. (2015). Simdatamapper: An architectural pattern to integrate declarative similarity matching into database applications. In *Proc. of the SBBD Conference*, pages 967–972.

Sidney, C. F., Mendes, D. S., Ribeiro, L. A., and Härder, T. (2015). Performance prediction for set similarity joins. In *Proc. of the the ACM Symposium on Applied Computing*, pages 967–972.

Wang, J., Li, G., and Feng, J. (2012). Can we beat the prefix filtering?: an adaptive framework for similarity join and search. In *Proc. of the SIGMOD Conference*, pages 85–96.

Xiao, C., Wang, W., Lin, X., and Yu, J. X. (2008). Efficient similarity joins for near duplicate detection. In *Proc. of the 17th Intl. Conf. on World Wide Web*, pages 131–140.

Efficient Self-similarity Range Wide-joins Fostering Near-duplicate Image Detection in Emergency Scenarios

Luiz Olmes Carvalho, Lucio F. D. Santos, Willian D. Oliveira, Agma J. M. Traina, Caetano Traina Jr.
Institute of Mathematics and Computer Sciences, University of São Paulo, São Paulo, Brazil
{*olmes, luciodb, willian, agma, caetano*}*@icmc.usp.br*

Keywords: Similarity Search, Similarity Join, Query Operators, Wide-join, Near-duplicate Detection.

Abstract: Crowdsourcing information is being increasingly employed to improve and support decision making in emergency situations. However, the gathered records quickly become too similar among themselves and handling several similar reports does not add valuable knowledge to assist the helping personnel at the control center in their decision making tasks. The usual approaches to detect and handle the so-called near-duplicate data rely on costly twofold processing. Aimed at reducing the cost and also improving the ability of duplication detection, we developed a framework model based on the similarity wide-join database operator. We extended the wide-join definition empowering it to surpass its restrictions and accomplish the near-duplicate task too. In this paper, we also provide an efficient algorithm based on pivots that speeds up the entire process, which enables retrieving the top similar elements in a single-pass processing. Experiments using real datasets show that our framework is up to three orders of magnitude faster than the competing techniques in the literature, whereas also improving the quality of the result in about 35 percent.

1 INTRODUCTION

Emergency situations can threaten life, environment and properties. Thus, a great effort are being made to develop systems aimed at reducing injuries and financial losses in crises situations. Existing solutions employ ultraviolet, infrared sensors and surveillance cameras (Chino et al., 2015). The problem of using sensors is that they need to be installed near to the prospected emergency places, and forecasting all the possible crisis situations in a particular region is not feasible.

On the other hand, surveillance cameras can provide visual information of wider spaces. When associated the increasing popularity of smartphones with good quality cameras and other mobile devices, they may lead to better solutions to map crisis scenarios and allow speeding up planning emergency actions to reduce losses. Seizing the opportunity to take such information into accont, the Rescuer[1] Project is developing an emergency-response system to assist Crisis Control Committees during a crisis situation. It provides tools that allow witnesses, victims and the rescue staff to gather emergency information based on images and videos sent from the incident place using a mobile crowdsourcing framework.

Crowdsourcing data can provide a large amount of information about the emergency scenario, but it often leads to a large amount of records very similar among themselves too. For instance, let us consider the occurrence of an event such as a building on fire or a serious incident in an industrial plant. As the eyewitnesses register the event with their smartphones many and repeatedly times, several pictures become copies almost identical to others. In the image retrieval context, the images too similar to each other with only smooth variations imposed by the devices or the capture conditions (resolution, illumination, cropping, rotation, framing) are called *near-duplicates* (Li et al., 2015; Yao et al., 2015).

For instance, Fig. 1 depicts a 9 days-long fire occurred in an industrial plant at Santos, Brazil, in April 2015. As shown in Fig. 1, eyewitnesses e_1, e_2 and e_3 took photos from the same perspective of the burning industrial plant, whereas e_4, e_5 and e_6 took photos from another side of the scenario and the same happened to eyewitness e_7, e_8, e_9 and e_{10}. Too much similar images from the same perspectives (near-duplicates) may not improve the decision making support. In this example, each image subset $\{img_1, img_2, img_3\}$, $\{img_4, img_5, img_6\}$, $\{img_7, img_8, img_9\}$ and

[1]Rescuer: Reliable and Smart Crowdsourcing Solution for Emergency and Crisis Management - <http://www.rescuer-project.org>

Carvalho, L., Santos, L., Oliveira, W., Traina, A. and Jr., C.
Efficient Self-similarity Range Wide-joins Fostering Near-duplicate Image Detection in Emergency Scenarios.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 81-91
ISBN: 978-989-758-187-8

$\{img_{10}\}$ forms near-duplicates. Thus, it is more useful *to remove* the near-duplicates, fostering more diversified results, which present a holistic vision about the incident, such as using only the subset $\{img_1, img_6, img_8, img_{10}\}$.

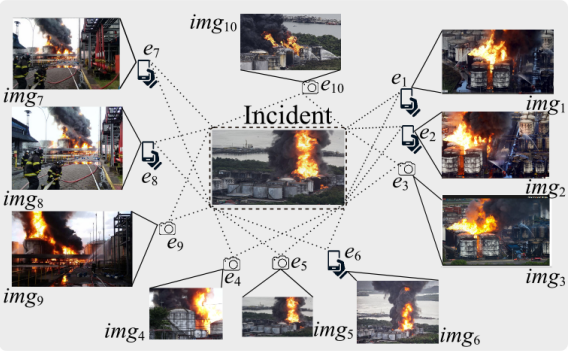


Figure 1: An example of taking photos of an emergency scenario (industrial plant on fire).

On the other hand, sometimes it is interesting to *retrieve* and *return* the near-duplicate elements. In the aforestated example (Fig. 1), law enforcement officers and investigators may be interested on the near-duplicate images. Although the near-duplicate elements may be not useful for non-expert people, those professionals usually see things differently and are trained to recognize small details among the images that can contribute to the investigation.

Near-duplicate image detection has attracted considerable attention in multimedia and database communities (Xiao et al., 2011; Wang et al., 2012; Li et al., 2015; Yao et al., 2015). However, to the best of our knowledge, the near-duplicate detection is yet an open problem, with no consolidated technique able to accomplish such task in terms of both efficiency and efficacy. Most of the approaches to detect near-duplicate images rely on executing a twofold processing, described as follows:

1. *Building*: the first phase aims at retrieving a candidate set of near-duplicate elements. It can use every individual image in the dataset as a similarity query center to retrieve the images most similar to each one (Xiao et al., 2011), or employ clustering-based techniques (Li et al., 2015).
2. *Improvement*: the second phase processes the candidate set and refines it removing false positives. The main differences among the distinct methods are in this phase. Most of them sacrifices the computational efficiency to enhance the result efficacy.

Considering the scenario depicted in Fig. 1, it is reasonable to consider that a crowdsourcing framework can be "flooded" with a large amount of images. In that case, employing a twofold technique may take too much time to generate the first perspective about the incident scene and, when it is achieved, the situation may already be changed.

As a possible solution, recent approaches (Xiao et al., 2011; Carvalho et al., 2015) consider using well-known searching operators from both the database and information retrieval areas, namely *similarity joins* and *wide-joins*, to detect near-duplicate elements. Similarity joins (Silva et al., 2013) obtain pairs of elements similar among themselves, assuming each pair corresponds to the near-duplicate candidates obtained by the building phase. However, those retrieval operators were applied only to detect near-duplicate elements in data represented by strings, as in (Xiao et al., 2011), but they were not explored in other domains such as images. In their turn, wide-joins (Carvalho et al., 2015) are designed to retrieve the overall most similar pairs, leading to an inherent combination of building and improvement phases in their processing. However, wide-joins process two distinct sets, while near-duplicate detection must combine a set with itself.

Although employing the join-based techniques may improve the performance, they require computing the similarity among all possible pairs of image received. As the amount of elements is usually large, the situation becomes similar to the first alternative. Therefore, both alternatives present drawbacks when applied to emergency control systems.

To detect near-duplicate images for emergency scenarios efficiently, this paper introduces a framework based on the similarity range wide-join database operator. We extended the operator definition to enable processing a single relation, thus we enlarged its usability as a unary *self wide-join* operator. Moreover, we devised an optimized algorithm based on pivots to speed up processing similarity wide-join, prioritizing early result generation, as required by emergency-based support systems, with no need of further improvement steps. Experiments performed on two real datasets show that our proposal is at least two orders of magnitude faster than existing techniques, whereas always returning a high-quality answer.

The remainder of this paper is organized as follows: Section 2 describes the main concepts and related work. Section 3 introduces our framework to detect near-duplicate images, the definition and algorithms for the self wide-join. Section 4 presents experimental evaluation of our technique and discusses the main results. Finally, Section 5 summarizes the main achievements and outlines future steps.

2 BACKGROUND

This Section overviews the main concepts and the related work to ours regarding to the image representation (Section 2.1), near-duplicates object detection (Section 2.2) and the evaluation of similarity queries, including the types similarity joins (Section 2.3). Also, the main symbols employed along the paper are summarized in Table 1.

2.1 Feature Extraction and Image Representation

Aiming at enabling retrieval by content and hence the near-duplicate detection, images are compared according to a similarity measure. To evaluate the similarity, images are represented by an n-dimensional array of numerical values, called *feature vector*, that describes their content. The features are numerical measurements of visual properties.

The algorithms responsible for processing images and obtaining their features are known as *feature extractors methods*. For each data domain there are specific features to be considered and, in the case of images, the *off-the-shelf* extractors capture features based on colors (e.g. histograms), texture (e.g. Haralick features) and/or shape (e.g. Zernike Moments) (Sonka et al., 2014).

The evaluation of the similarity measure between two feature vectors is performed by a *metric*. Formally, given a feature vector space $\mathbb{D}$ (the data domain), a distance function $d : \mathbb{D} \times \mathbb{D} \mapsto \mathbb{R}^+$ is called a *metric* on $\mathbb{D}$ if, for all $x, y, z \in \mathbb{D}$, there holds:

- $d(x,y) \geq 0$ (non-negativity)
- $d(x,y) = 0 \Rightarrow x = y$ (identity of indiscernibles)
- $d(x,y) = d(y,x)$ (symmetry)
- $d(x,y) \leq d(x,z) + d(z,y)$ (triangle inequality)

The pair $\langle \mathbb{D}, d \rangle$ is called *metric space* (Searcóid, 2007). The metric space is the mathematical model that enables to perform similarity queries, and hence to detect the near-duplicate elements.

2.2 Near-duplicate Detection

Several techniques for near-duplicate detection rely on the Bag-of-Visual Words (BoVW) model (Li et al., 2015; Yao et al., 2015). That model represents the features as *visual words* and the image representation consists of counting the words to create an histogram. However, BoVW reliability for duplicate detection is small, as it does not capture the spatial relationship existing among the extracted features.

Aimed at surpassing this drawback, studies like (Yao et al., 2015) combined the spatial information with the BoVW local descriptors. However, local descriptors yet generate feature vectors of varying dimensionality, which is troublesome to represent in metric spaces, requiring high-costly metrics.

Other approaches considered to spot near-duplicates are hash functions and the Locality Sensitive Hashing (LSH) (Bangay and Lv, 2012; Wang et al., 2012). Whereas hash functions fail on representing information for similarity retrieval, once small differences in images leads to distinct hash representations, the LSH circumvents this problem by retrieving approximate result sets. In the same line, weighted min-Hash functions improve the image representation, but once they are usually based on bag-of-words, they may present the same drawbacks of the other technique (Chum et al., 2008). Unlike those techniques, we are interested in accurate answers.

Still worth to mention is the "Adaptive Cluster with k-means" technique (ACMe) (Li et al., 2015). It applies clustering algorithms to group near-duplicate images in a twofold process. First it clusters the dataset using the k-means algorithm. Subsequently, the coherences of the obtained clusters are checked to determine the need of recursively processing each cluster. The result is then refined using local descriptors. This is a highly expensive technique that requires for the Improvement phase. Moreover, as the k-means algorithm is sensitive to outliers, our intuition is that better quality result might be achieved replacing it with the k-medoids algorithm. Surpassing the existing drawbacks in algorithms that have an improvement phase, our proposal extends similarity joins to speed up the detection of near duplicates without post-processing the image database.

2.3 Similarity Join

Similarity joins are database operators that combine the tuples of two relations T_1 and T_2 so that each retrieved pair $\langle t_1 \in \mathsf{T}_1, t_2 \in \mathsf{T}_2 \rangle$ satisfies a similarity predicate θ_s. The similarity conditions most employed in similarity joins generate the similarity range join and the k-nearest neighbor join (Silva et al., 2013).

Assume that each relation has an attribute $\mathsf{S}_1 \subseteq \mathsf{T}_1$ and $\mathsf{S}_2 \subseteq \mathsf{T}_2$, both sampled from the same metric space $\langle \mathbb{D}, d \rangle$. Given a maximum similarity threshold ξ, the *similarity range join* retrieves the pairs $\langle t_1, t_2 \rangle$, $t_1 \in \mathsf{T}_1$ and $t_2 \in \mathsf{T}_2$, such that $d(t_1[\mathsf{S}_1], t_2[\mathsf{S}_2]) \leq \xi$. Given an integer value $k \geq 1$, the *k-nearest neighbor join* retrieves $k * |\mathsf{T}_1|$ tuples $\langle t_1, t_2 \rangle$ such that t_2 is one

Table 1: Symbols.

SYMBOL	MEANING
ξ	similarity limiar
$\mathbb{D}$	data domain
d	distance function / metric
$\mathfrak{F}$	feature extractor method
f	feature value
img	an image
k, κ, m, n	integer values
$\mathsf{S}, \mathsf{S}_1, \mathsf{S}_2$	attributes subject to a metric
$\mathsf{T}, \mathsf{T}_1, \mathsf{T}_2$	relations
t, t_1, t_2, t_i, t_j	tuples
$t[\mathsf{S}]$	the value of attribute S in tuple t
v	feature vector

of the k most similar attributes to each t_1 (Carvalho et al., 2015).

A third type of similarity join is often described in the database literature (Silva et al., 2013): the k-distance join. It retrieves the k pairs $\langle t_1, t_2 \rangle$ having the most similar values $t_1[\mathsf{S}_1]$ and $t_2[\mathsf{S}_2]$. This operator is an instance of the similarity *wide-join* (Carvalho et al., 2015). Wide-joins retrieve the most similar pairs in general, sorting the tuple internally, allowing its processing to comply with the relational theory and executed efficiently.

Similarity joins can be used to perform several tasks, including near-duplicate detection. For this last purpose, however, similarity joins have been explored only in string-based data represented as tokens, using metrics such as the Edit or Hamming distance, as in (Xiao et al., 2011). Our proposal considers other domains but string data and employs more general metrics, such as the Minkowski (L_p) family over image domains. Likewise, similarity wide-joins have been restricted used to operate on two distinct relations, loosing optimization opportunities that exists when processing elements lying in the same set.

3 NEAR-DUPLICATE DETECTION

Detecting near-duplicates on multimedia repositories plays an important role in presenting a more useful result, as returning images too much similar not only poses a negative impact on the retrieval time, but generally it also reduces the users' browsing experience. Imposing users to interactively analyze near-duplicates until obtaining the desired result is annoying, and requires a lot of time that would be more wisely employed specially when handling emergency scenarios. Following, we present a novel framework to detect near-duplicates (Section 3.1) and an extension of the similarity wide-join definition, which greatly reduces such drawbacks (Section 3.2). Last but not least, Section 3.3 presents the basic approach to implement our proposed wide-join extension and also devises an optimized version based on pivots to achieve an efficient computation.

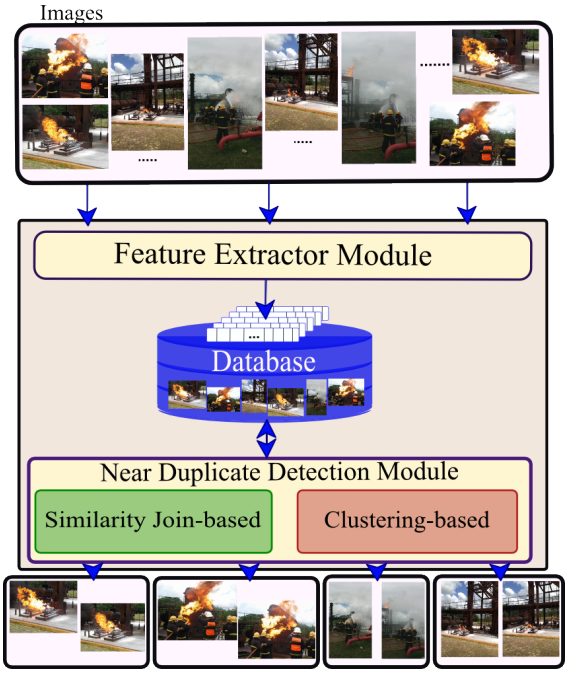


Figure 2: The architecture of the framework for near-duplicate detection.

3.1 The Framework Architecture

The proposed framework is composed of two modules, organized according to Figure 2. The Feature Extractor module processes images such as a content-based image retrieval system, representing them as n-dimensional arrays. Formally, the Feature Extractor module receives an image repository $C = \{img_1, \ldots, img_m\}$ with m images, and extracts the visual features $v_i = \mathfrak{F}(img_i)$ of each image $img_i \in C$. Each v_i is a feature vector $\langle f_1, \ldots, f_n \rangle$, where n is the number of features extracted by the feature extractor method $\mathfrak{F}$. The features depend on the kind of visual aspect considered, e.g., color, shape, texture, etc, as discussed in Section 2.1. Its result is m Feature Vectors stored into the S attribute in relation T such that $\mathsf{T}[\mathsf{S}] = \{v_1, \ldots, v_m\}$, and the corresponding images are stored as another attribute in T.

The second module – Near Duplicate Detection – is the core module of our framework. It can perform either a Similarity Join-based or a Clustering-based

near-duplicate detection. Both compare the feature vectors according to a distance function. The detection based on similarity join executes our specialized similarity join operator (described in Section 3.2) on T, employing two user-defined parameters that allows tuning the comparison to follow the user's perception of how pairs of images can be considered as near-duplicates. The Cluster-based detection processes T executing one of the defined clustering techniques: the Adaptive Cluster with *k*-means (ACMe - Section 2) or our Adaptive Cluster with *k*-medoids variation (ACMd - Section 4). The framework returns the resulting pairs of images that each algorithm detects as near-duplicates, allowing comparing the considered techniques. Thus, our proposal allows users either removing, preserving or return the near-duplicate according to the interest of users.

3.2 Self Similarity Wide-joins

The similarity joins operators, such as the range join and the *k*-nearest neighbor join, present shortcomings when employed to query databases. Both of them return result sets whose cardinality is often too high, which leads to many more pairs of elements than users really need or expect. Hence, that large result set usually includes pairs truly similar, as well as pairs holding a low or even questionable degree of similarity. Therefore, to fulfill the near-duplicate task, the result of a similarity join must be further processed in order to exclude the pairs whose the similarity measure is doubtful.

Moreover, the *k*-nearest neighbor join is also troublesome because it does not assure an equivalent similarity among the *k*-th nearest pairs from distinct elements. Thus, given two vectors $v_i = t_i[\mathsf{S}]$ and $v_j = t_j[\mathsf{S}]$, the distances ξ_i and ξ_j from v_i to its *k*-nearest neighbor, let v_{ik}, and from v_j to its *k*-nearest neighbor, let v_{jk}, are completely uncorrelated. In this way, for any given $k \geq 1$, a pair $\langle v_i, v_{ik} \rangle$ may be a near-duplicate whereas the pair $\langle v_j, v_{jk} \rangle$ may not. Hence, looking at the range ξ variation in the *k*-neighbors becomes the main focus of our investigation.

Our proposal is that the resulting pairs of a similarity range join must have the similarity between their component elements evaluated and subsequently ranked so that the top-ranked ones correspond to the near-duplicate elements. Such kind of processing can be efficiently achieved by extending the similarity join operator called range wide-join (Section 2).

Wide-joins are intended to compute the similarity join between two relations and retain only the global most similar elements. The near-duplicate detection requires combining a set with itself, but wide-joins do not comply with such processing once the most similar pairs will include combinations of each element with itself, distorting the result.

For this purpose, we employ a tailored version of the wide join operator, namely *self range wide-join*, that atomically performs (i) the similarity evaluation over the *same set or relation* and (ii) the retrieval of the most similar elements in general. Those two operations intrinsically coupled as a single operator enable retrieving the element pairs considered as near-duplicates in a single-pass, avoiding further processing of refinement phase.

Formally, let $\mathbb{D}$ be a data domain, $d : \mathbb{D} \times \mathbb{D} \mapsto \mathbb{R}^+$ be a metric over $\mathbb{D}$, T be a relation, $\mathsf{S} \subseteq \mathsf{T}$ be an attribute subject to d with values sampled from $\mathbb{D}$, ξ be a maximum similarity threshold and κ be an upper bound integer value. The *self similarity range wide-join* is given by Definition 1.

Definition 1 (The Self similarity range wide-join)**.** *The self similarity range wide-join* $\boxed{\bowtie_{(\mathsf{S},\xi,\kappa)}\mathsf{T}}$ *is a similarity range join where both left and right input relations* T_1 *and* T_2 *are the same relation* T*, and it returns at most* κ *pairs* $\langle t_1, t_2 \rangle | t_1, t_2 \in \mathsf{T}$ *such that* $t_1 \neq t_2$*,* $d(t_1[\mathsf{S}_1], t_2[\mathsf{S}_2]) \leq \xi$ *and the returned pairs are the* κ *closest to each other. The self range wide-join is expressed in relational algebra according to* (1).

$$\bowtie_{(\mathsf{S},\xi,\kappa)}\mathsf{T} \equiv$$
$$\pi_{\{\mathsf{T}_1,\mathsf{T}_2\}}\Big(\sigma_{(ord\leq\kappa)}\Big(\pi_{\{\mathsf{T}_1,\mathsf{T}_2,\mathcal{F}(d(t_1[\mathsf{S}_1],t_2[\mathsf{S}_2]))\rightarrow ord\}}\Big(\rho_{(\mathsf{S}/\mathsf{S}_1)}(\mathsf{T}/\mathsf{T}_1) \overset{d(t_1[\mathsf{S}_1],t_2[\mathsf{S}_2])\leq\xi}{\bowtie} \rho_{(\mathsf{S}/\mathsf{S}_2)}(\mathsf{T}/\mathsf{T}_2)\Big)\Big)\Big) \quad (1)$$

The self similarity range wide-join is a *unary* operator (it takes one relation) that internally performs a range join, sorts the intermediate result by the dissimilarity among the tuples and returns the top-κ pairs $\langle t_i, t_j \rangle$ of most similar elements in T. In (1), $\mathcal{F}$ is a database aggregate function that receives the distances between the attributes $t_1[\mathsf{S}_1]$ and $t_2[\mathsf{S}_2]$ and projects the ordinal classification of those dissimilarity values into an attribute *ord* that exists only during the operator execution. Further, that transient attribute is used to select the most similar pairs and discarded.

Following (1), the self similarity range join relies on the maximum limiar ξ in order to filter the candidate pairs to compose the answer. This operation is related to the building processing phase, where two images *a* and *b* are *possible* near-duplicates *iff* the dissimilarity between them is at most the threshold ξ, that is, $d(a,b) \leq \xi$.

The inner similarity join may be influenced by the data distribution. Each attribute S_1 of the pairs $\langle t_1, t_2 \rangle$

Algorithm 1: NLWJ(T, ξ, κ).

```
1  Q ← ∅;
2  for i ← 1 to |T| − 1 do
3      for j ← i + 1 to |T| do
4          dist ← d(t_i[S], t_j[S]);
5          if dist ≤ ξ then
6              if |Q| ≤ κ then
7                  Q ← Q ∪ {⟨⟨t_i, t_j⟩, dist⟩};
8              else
9                  Let q ∈ Q be the high-priority
                     element;
10                 if dist < d(q[S_1], q[S_2]) then
11                     Q ← Q − {q};
12                     Q ← Q ∪ {⟨⟨t_i, t_j⟩, dist⟩};
13 return Q;
```

can be combined with varying quantities of values in S_2. Thus, the inner join in (1) retrieves pairs in a large range of distances, but only the smaller distances truly correspond to near-duplicate elements. The greater the distance among S_1 and S_2, the smaller the confidence that the pair is a near-duplicate.

Sorting the self-similarity of the pairs is related to the improvement phase of a near-duplicate process. As it is performed internally by the wide-join, no further processing is required. In addition, that step also solves a frequent issue existing in traditional similarity joins: how to define ξ. Once the self range wide-join sorts the pairs and filters just the closest, the ξ parameter can be overestimated without adversely affecting the quality of the final answer. Moreover, it improves both the query answer quality and the performance of the self similarity range wide-join operator, as it was confirmed by the experiments reported in Section 4.

3.3 Algorithmic Issues

Self similarity range wide-joins can be implemented more efficiently than the sequence expressed in (1), following a strategy based on a nested-loop (Nested-Loop Wide-Join - NLWJ), as depicted in Algorithm 1. Usually, the traditional range wide-join performs n^2 distance computations, where $n = |T|$. However, our self version of the algorithm (steps 2 and 3) requires only half of that amount because, as the join condition is a metric, it meets the symmetry property. Therefore, a first improvement is that it is necessary to compute the distances $d(t_i[S], t_j[S])$ and $d(t_j[S], t_i[S])$ only once, resulting in $n(n-1)/2$ distance calculations.

Following, the κ most similar pairs qualifying as near-duplicates (steps 5-6) are added into a priority queue Q (step 7). The priority parameter is the similarity distance: a greater distance corresponds to a higher priority for removal. After κ pairs were obtained, the current pair $\langle t_i, t_j \rangle$ replaces the higher priority element (q - step 9) whenever it is more similar than q, as tested in step 10. Thus, the second improvement is truncating the sorting operation, as $\mathcal{F}$ in (1) can be incrementally performed by the priority queue, which avoids the cost of sorting the total whole amount of pairs or overflowing memory with too many elements. When the procedure finishes, the priority queue Q already contains the near-duplicate images.

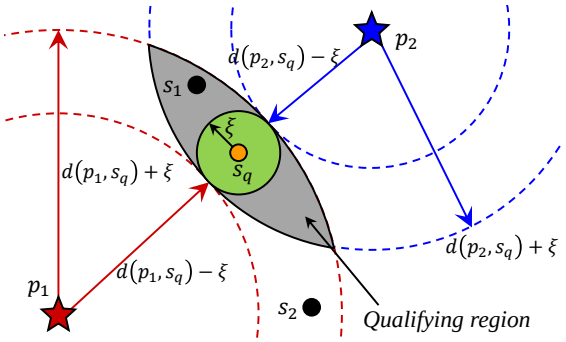


Figure 3: Pivot-based strategy to prune the search space.

Finally, we designed a third improvement to compute self-similarity range wide-joins. The trick here is based on the triangle inequality property of a metric (Section 2), using pivot elements in order to prune the in-list search space and further reduce the number of distance computations.

For an arbitrary and small number $p \ll n$ of pivots chosen among the available element in the database, we first compute the distance between each element $t_i[S]$ to each one of the p pivots. In such manner, each element in the database is filtered by their distances to each pivot. Notice that, until this step, only $p * n$ distance computations were performed.

The next step performs the similarity join. Each element s_q is compared to each elements s_i, and when they are closer than the similarity threshold ξ, the pair is part of the answer. To prune comparisons, it is first verified if s_i is within the qualifying area of s_q regarding to each pivot, assuring that for each pivot p_i the conditions defined in (2) and (3) simultaneously hold.

$$d(p_i, s_i) \geq d(p_i, s_q) - \xi \quad (2)$$

$$d(p_i, s_i) \leq d(p_i, s_q) + \xi \quad (3)$$

For instance, element s_2 in Fig. 3 satisfies conditions (2) and (3) with respect to the pivot p_1, i.e., s_2 is within the hyper-ring delimited by the pivot p_1. However, when analyzed in relation to the pivot p_2, s_2 does not satisfy (3), thus it is guaranteed that s_2 is out-

side the intersection area among the two hyper-rings. Therefore, the comparison of s_q with s_2 is pruned.

Still considering Fig. 3, notice that the element s_1 holds conditions (2) and (3) with respect to both pivots and should be compared to s_q. In this case, although s_1 is within the qualifying area defined by the two pivots, it is not within the range area of s_q, which can be verified with just one distance computation.

For those elements within the qualifying area, the number of additional distance computations is based on the data distribution and cannot be predicted beforehand. However, the worst and highly improvable situation occurs when the n elements in the database lie within the qualifying area. In this case, it is necessary to perform, for each element s_q, n distance computations, which leads to a total of $np + n(n-1)/2$ calculations. Similar to the nested-loop case, due to the symmetry property of the metric (Section 2), at most a half of all the distance computations are required.

Nevertheless, it is very uncommon and easy to avoid to have all the dataset in the qualifying area. In Fig. 3, notice that making $p = 3$, thus putting a third pivot next to the element s_2, would substantially reduce the qualifying area, restricting it to almost the coverage region of s_q.

The opposite situation occurs when no element qualifies. In that case, there is no distance computation. In average, the number of distance computations can be estimated as the arithmetic mean among the best and worst cases, which leads the required number of distance calculations to be significantly less than $n(n-1+2p)/4$ distance computations, already including the $n * p$ pivot-elements performed calculations.

Similarity wide-join based on pivots (WJ-P) can be implemented in external memory following the block-nested loop approach introduced in Algorithm 2. Similar to Algorithm 1, the WJ-P implementation also relies on a priority queue Q (step 1) in order to achieve the sorting step of similarity wide-joins. In step 2, p pivots are chosen at random. Heuristics on how the pivots should be chosen are out of scope in this paper. Steps 3-6 iterate over the blocks where the tuples are stored. The nested-loop of steps 8 and 12 iterates over the elements inside the blocks. In order to avoid combining a element with itself ensuring a self similarity join, the condition in step 10 increments the start position of the inner loop.

For each pivot picked in step 2, Equations (2) and (3) must hold (step 13). Notice that the distance between the elements and the pivots can be precomputed and stored when reading the elements in the loops of steps 8-12. Step 13 means that the analyzed tuple (t_y)

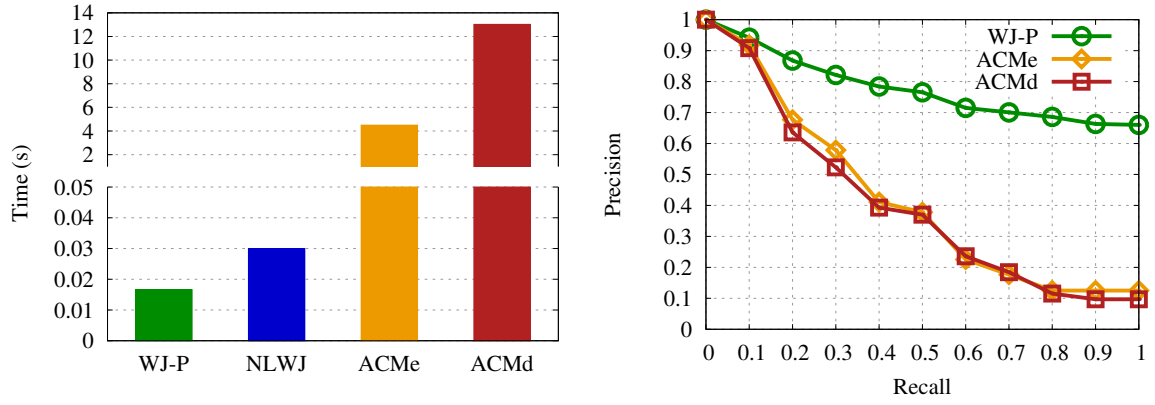
Algorithm 2: WJ-P(T, ξ, κ).

```
1  Q ← ∅;
2  Choose p pivots at random in T;
3  for i ← 1 to number of blocks of T do
4      for j ← i+1 to number of blocks of T do
5          load block i to memory;
6          load block j to memory;
7          x ← 1;
8          while x < number of elements in block i
               do
9              y ← x;
10             if i = j then
11                 y ← y+1;
12             while y < number of elements in
                   block j do
13                 if expressions (2) and (3) hold
                       ∀ pivot p then
14                     dist ← d(t_x[S], t_y[S]);
15                     if dist ≤ ξ then
16                         if |Q| ≤ κ then
17                             Q ← Q ∪
                                   {⟨⟨t_x, t_y⟩, dist⟩};
18                         else
19                             Let q ∈ Q be the
                                   high-priority
                                   element;
20                             if dist <
                                   d(q[S_1], q[S_2])
                                   then
21                                 Q ← Q − {q};
22                                 Q ← Q ∪
                                       {⟨⟨t_x, t_y⟩, dist⟩};
23 return Q;
```

is within the qualifying hyper-ring defined by the pivots. The pertinence of t_y to the region convered by t_x is then checked in step 14-15, where an additional distance computation was performed. The steps 15-22 are similar to those presented in Algorithm 1, where κ elements are selected so the algorithm checks for possible replacements of the most similar pairs.

4 EXPERIMENTS

This section reports on experiments using our framework for near-duplicate image detection. The goal is to evaluate the proposed self range wide-join tech-



(a) Runtime

(b) Precision × Recall

Figure 4: Performance and quality analysis: the `Fire` dataset.

nique in terms of both the computational performance and the answer quality, targeting prioritizing those aspects as required for an emergency monitoring system.

4.1 Experiment Setup

We describe the results performed on a real dataset - `Fire` - composed of 272 images of fire incidents. Those images were obtained from an emergency situation simulation, held in an Industrial Complex. The dataset was previously labeled by domain experts and 25 distinct incident scenes were recognized. The images were submitted to the Color Layout (Kasutani and Yamada, 2001) extractor, which generated 16 features. The L_2 (Euclidean) metric was employed to evaluate the vector distances.

We compared our improved pivot-based self range wide-join (WJ-P, Algorithm 2) using 5 pivots with three other techniques. The first is the similarity wide-join with a nested-loop approach (NLWJ), that is the *self version* of the baseline found in the literature (Carvalho et al., 2015). The second method is the Adaptive Cluster with k-means (ACMe - Section 2) (Li et al., 2015). Also, once k-means is sensitive to outliers and often computes "means" that do not correspond to real dataset images, we generated a third method that is an ACMe variant replacing k-means with the k-medoids algorithm, calling it ACMd, in order to better analyze the answer quality. When necessary, the parameter ξ was set to retrieve about 1% of the total number of possible pairs.

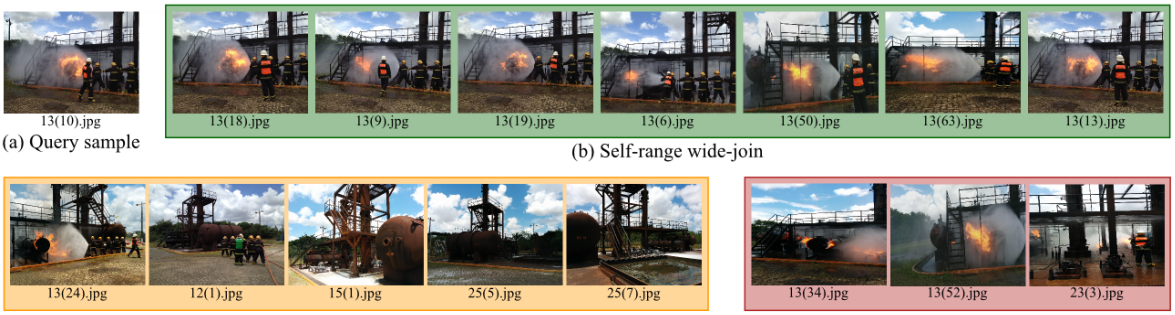
The experiments were executed in a computer with an Intel® Core™ i7-4770 processor, running at 3.4 GHz, with 16 GB of RAM under Ubuntu 14.04. All evaluated methods were implemented in C++. Each technique was evaluated with respect to both the total running time (Section 4.2) and the answer quality (Section 4.3), as follows.

4.2 Performance Experiment

Fig. 4(a) presents the total running time of the four approaches evaluated. The reported time corresponds just to the execution of the Near Duplicate Detection Module, as feature extraction was performed only once to provide data to the four methods. In this experiment, WJ-P was 44.45% faster than NLWJ. Also, both techniques based on self wide-join were 2 orders of magnitude faster than ACMe and 3 orders of magnitude faster than ACMd, whereas returning a high quality result set.

Such behavior occurs due to the fact that ACMe and ACMd cluster the dataset and recursively redistribute the elements following a hierarchical approach for the improvement phase, until the coherence of each cluster does not exceed a maximum value, computed during the process. In addition, to achieve the result, an improvement phase is usually required, which contributes to increase the computational cost of those approaches. Distinctly, the WJ-P performs a single pass computation that embodies the building and the improvement phases into an atomic, optimized operation.

Both ACMe and ACMd require a parameter k to execute their core clustering algorithms, the k-means and k-medoids, respectively. Nevertheless, they returned a number of clusters greater than k, because the clusters obtained in the building phase are subdivided according to their coherence values. Those techniques achieved the better results when k is set to values between 20 and 30. Unlike, the WJ-P method was able to achieve the result without the need of several executions in order to find out the better parameter adjustments.



(a) Query sample (b) Self-range wide-join

(c) Adaptive cluster with *k*-means (d) Adaptive cluster with *k*-medoids

Figure 5: Near-duplicates obtained by the three evaluated methods for the query center shown.

4.3 Answer Quality Evaluation

To analyze the answer quality, we evaluated how accurate is the result returned by the proposed framework. In order to enable fair comparisons among the distinct algorithms, we computed Precision and Recall (P $\times$ R) curves. Evaluating P $\times$ R is a common technique used for information retrieval evaluation. Precision is defined as the rate of the number of relevant elements retrieved by the number of retrieved elements. Recall is given as the rate between the number of relevant elements retrieved by the number of relevant elements in the database. In P $\times$ R plots, the closer a curve to the top, the better the corresponding method.

Fig. 4(b) shows the P $\times$ R curves achieved by the three approaches. In this experiment, we did not consider the NLWJ approach because its result is the same also produced by the WJ-P and therefore both curves are identical, only varying their runtime. Our self range wide-join based on pivots achieved the larger precision for every recall amount. It was, in average, 35.14% more precise than ACMe and 36.78% than ACMd. After retrieving all relevant images in the dataset (recall of 100%), WJ-P consistently obtained 66.00% of precision in the result, whereas the competitor techniques achieved a maximum precision of 12.50%.

In order to show the obtained gain of precision, Fig. 5 samples the images considered as near-duplicates by the three techniques. Again, NLWJ is omitted once it computes the same result of the WJ-P, but the former is slower than the latter. For an image randomly chosen as query center (Fig. 5(a) - the 10th image with label 13), Fig. 5(b) shows the near-duplicates retrieved by WJ-P. As it can be seen, they have the same label and are in fact related to the query, recognizing even images with zoom and rotations.

Figs. 5(c) and 5(d) show the clusters obtained by the ACMe and ACMd, respectively. Both are the clusters where the query image (Fig. 5(a)) was allocated As it can be noted, both methods retrieved false positives, where the existence of false positives contributed to decrease the precision of ACMe and ACMd. Although ACMe theoretically leads to worse clusters than ACMd, as the means are not real images whereas the medoids are, the precision difference among both was in average only 1.63% (see Fig. 4(b)).

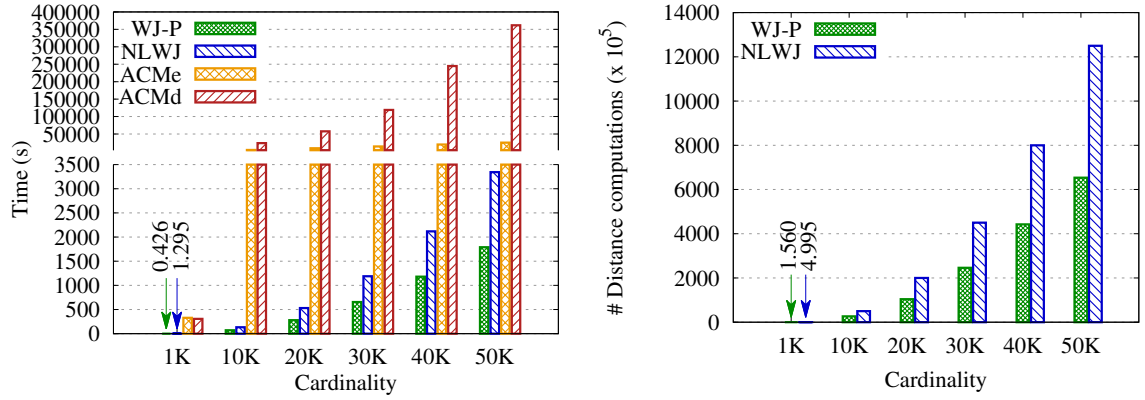
The superior quality of the answer of WJ-P when compared to the cluster-based methods shown in Fig. 4(b) is explained by the fact that the clusters are generated based on centroids or medoids seeds, and the remaining elements are allocated according to their distances to the seeds. The cluster elements are analyzed only in relation to the seeds, ignoring the relationship among themselves. This fact leads to some images that are distinct among them but considered similar to their seed, as represented in Figs. 5(c) and 5(d). In its turn, the self wide-join method establishes a "pairing relationship" among the elements, avoiding such drawback and increasing the answer quality, as also observed in Fig. 5(b).

Notice that Fig. 5 shows the images *spoted as near-duplicates* by each of the three techniques. According to the user interest, those near-duplicates can be either removed from the final answer of the framework so as to provide a more informative result set, or returned, allowing to analyze similar occurrences.

4.4 Scalability Analysis

The `Fire` dataset contains real images from an emergence scenario from the Rescuer Project, but it contains few images. So, we evaluated the scalability of our technique employing the `Aloi` dataset[2]. It contains images of 1,000 objects rotated from 0º to 360º in steps of 5º(72 images per object, which we assume to be near-duplicates) giving a total of 72,000 distinct images. The Color Moment extractor (Stricker and

[2] <http://aloi.science.uva.nl> Access: Sept. 11, 2015.



(a) Runtime (b) Distance computations

Figure 6: Scalability analysis: `Aloi` dataset.

Orengo, 1995) generates 144 features, which were compared using the L_2 metric.

For the scalability evaluation, we shuffled the `Aloi` images and varied the cardinality of the submitted data in several executions of our framework. Fig. 6(a) depicts the total runtime of each algorithm. The pivot-based self range wide-join (WJ-P) was in average 49.58% faster than NLWJ. Also, it was 96.25% faster than ACMe and 99.59% than ACMd, that is, it was correspondingly 2 and 3 orders of magnitude faster. For example, for a cardinality of 10,000 objects, WJ-P execution took 1.16 minutes and NLWJ took 2.20 minutes, while ACMe took 1.18 hours and AMCd took 6.61 hours.

Finally, Figure 6(b) compares both the similarity wide-join techniques (NLWJ and WJ-P) with respect to the number of distance computations performed to achieve the result. The pivot-based self range wide-join (WJ-P) executed at least 44.69% less distance calculations than NLWJ for a cardinality of 40K, but the greatest reduction of distance computations was observed for a cardinality of 1K, where WJ-P performed 68.75% less calculations. Nevertheless, it is important to highlight that both techniques obtained the same final result, but WJ-P was able to save computational resources in its processing.

Fig. 6 shows that as the cardinality increases, the cluster-based methods need to process more elements in the building phase. However, the coherence value is not used in this phase, so the next one (refinement phase) requires more iterations to subdivide the clusters and ensure that coherence is maintained. Distinctly, the wide-join perform a one-pass strategy. The increased cardinality turns the process more costlier, but in a less pronounced way as compared to the cluster-based ones. Moreover, to avoid performing distance calculations among every pair as occurs in NLWJ, the WJ-P prunes the number of comparisons, which also reduces the running time.

4.5 Experiment Highlights

In a general way, there are three main reasons explaining why the introduced self similarity range wide-join technique overcomes its correlates:

- *Single-pass computation*: usually, the near-duplicate detection is divided into two phases. The wide-join operator surpass the requirement for a refinement phase. As aforestated, such one-pass execution allowed to reduce the cost of the entire process in 2 orders of magnitude.
- *Efficient prune technique*: a prune technique based on pivots enables the proposed WJ-P algorithm to perform a reduced amount of element-to-element comparisons. The pivots delimit small regions of the space to be analyzed, allowing to discard several elements that surely will not compose the answer. Also, such strategy reduced the number of distance computations in about 44% in relation to the traditional nested-loop approach.
- *Similarity relationship between elements*: unlike the cluster methods, that computes the proximity of an element to a group, the self wide-join operator establishes a similarity relationship between each distinct pair of elements. It avoids the diversity found in two elements lying in opposite sides of a cluster, which increased the answer quality in about 35% in relation to the existing approaches.

5 CONCLUSIONS

In this paper we presented a framework model to detect near-duplicates using the similarity wide-join

database operator as its core. We introduced the self range wide-join operator: an improved version of the wide-join that enables computing similarity by combining a relation to itself. We optimized the wide-join algorithm to scan the search space relying on pivots and using metric space properties to prune elements, which enabled achieving a large performance gain when compared to the existing solutions.

The experiments were executed using two real datasets. They showed that our proposed wide-join-based framework is able not only to improve the near-duplicate detection performance by at least 2 and up to 3 orders of magnitude, but also to improve the quality of the results when compared to the previous techniques.

The introduced technique is general enough to be applied over any dataset in a metric space, but we focused its application for an emergency-based application. When handling an emergency scenario, it is common that the eyewitnesses capture a large amount of photos and videos about the incident. Existing monitoring systems can benefit from those crowd-sourcing information, aiming at improving decision making support. However, as the information increases, its elements tend to become too similar, so it is crucial to provide efficient techniques to properly handle near-duplicates.

As future work, we are exploring data distribution statistics and selectivity estimations for join operators in order to provide accurate definitions of the parameters required by the self-similarity range wide-join. We also intend to combine the images with their associated meta-data in order to further improve both the precision and the performance of near-duplicate detection.

ACKNOWLEDGEMENTS

The authors are grateful to FAPESP, CNPQ, CAPES and Rescuer (EU FP7-614154 / CNPQ 490084/2013-3) for their financial support.

REFERENCES

Bangay, S. and Lv, O. (2012). Evaluating locality sensitive hashing for matching partial image patches in a social media setting. *Journal of Multimedia*, 1(9):14–24.

Carvalho, L. O., Santos, L. F. D., Oliveira, W. D., Traina, A. J. M., and Traina Jr., C. (2015). Similarity joins and beyond: an extended set of operators with order. In *Proc. 8th Int. Conf. on Similarity Search and Applications*, pages 29–41.

Chino, D. Y. T., Avalhais, L. P. S., Rodrigues Jr., J. F., and Traina, A. J. M. (2015). Bowfire: detection of fire in still images by integrating pixel color and texture analysis. In *Proc. 28th Conf. on Graphics, Patterns and Images*, pages 1–8.

Chum, O., Philbin, J., and Zisserman, A. (2008). Near duplicate image detection: min-hash and tf-idf weighting. In *British Machine Vision Conference*, pages 1–10.

Kasutani, E. and Yamada, A. (2001). The mpeg-7 color layout descriptor: a compact image feature description for high-speed image/video segment retrieval. In *Proc. 8th Int. Conf. on Image Processing*, pages 674–677.

Li, J., Qian, X., Li, Q., Zhao, Y., Wang, L., and Tang, Y. Y. (2015). Mining near-duplicate image groups. *Multimedia Tools and Applications*, 74(2):655–669.

Searcóid, M. Ó. (2007). *Metric spaces*. Springer.

Silva, Y. N., Aref, W. G., Larson, P.-A., Pearson, S., and Ali, M. H. (2013). Similarity queries: their conceptual evaluation, transformations, and processing. *The VLDB Journal*, 22(3):395–420.

Sonka, M., Hlavac, V., and Boyle, R. (2014). *Image Processing, Analysis, and Machine Vision*. Cengage Learning.

Stricker, M. and Orengo, M. (1995). Similarity of color images. In *Proc. 3rd Conf. on Storage and Retrieval for Image and Video Databases*, pages 381–392.

Wang, X.-J., Zhang, L., and Ma, W.-Y. (2012). Duplicate search based image annotation using web-scale data. *Proc. of the IEEE*, 100(9):2705–2721.

Xiao, C., Wang, W., Lin, X., Yu, J. X., and Wang, G. (2011). Efficient similarity joins for near-duplicate detection. *ACM Transactions on Database Systems*, 36(3):15:1–15:41.

Yao, J., Yang, B., and Zhu, Q. (2015). Near-duplicate image retrieval based on contextual descriptor. *IEEE Signal Processing Letters*, 22(9):1404–1408.

SHORT PAPERS

Migration Results to a Private Cloud by using the M2CCF

Abílio Cardoso[1] and Fernando Moreira[1,2]
[1]*Institute for Legal Research and Departamento de Economia, Gestão e Informática, Universidade Portucalense, Rua Dr. António Bernardino de Almeida 541, 4200-072 Porto, Portugal*
[2]*IEETA, Universidade Aveiro, Campus Universitário de Santiago, 3810-193 Aveiro, Portugal*
{abilio.cardoso, fmoreira}@upt.pt

Keywords: Cloud Computing, Cloud Computing Adoption, ITIL.

Abstract: The cloud computing paradigm is transforming the way IT services are provided and consumed by changing IT products to services. The migration of in-house IT services to cloud computing must be performed carefully so as not to cause high losses in the institution. In this paper, we present the use of the framework developed by the same authors, to the migration of services, applications, data and infrastructures to cloud computing, M2CCF, compatible with Information Technology Infrastructure Library (ITIL). The work also discusses the results gathered from the real implementation of the framework in the migration of IT services to a private cloud.

1 INTRODUCTION

A growing number of organizations is expected to migrate their IT systems to cloud computing (CC) (Tušanová, 2012). Conversely, the migration to CC has a great growth potential with the current and predicted total budget to be spent on its services (Nkhoma and Dang, 2013). Indeed, there is few literature available on the process and methodological guidance on migrating existing software systems to cloud computing (Chauhan and Babar, 2011), namely because it is a new and evolving field (Conway and Curry, 2013).

The difficulties organizations are faced with when migrating their IT to CC, has started to gain the attention of the research community with works published on the topic such as (Ezzat, et al., 2011; Khajeh-Hosseini, et al., 2011; Kumar and Garg, 2012). However, none of these works has presented a systematic process, sufficiently detailed, in order to be useful as a guide for IT managers throughout the steps and decisions involved in a typical migration to CC. Moreover, this work was also triggered by the absolute need to improve research in CC as well as in IT Service Management (ITSM) identified by Robert Heininger (2012).

The migration to the CC paradigm by an organization requires a deep understanding of the institution IT as well as the dynamics of CC. By other side, there is already an extensive set of recommendations for IT management and IT governance in general such as the Information Technology Infrastructure Library (ITIL). Accordingly, we developed the Migration to Cloud Computing Framework (M2CCF) to support the migration to CC which utilizes the information gathered and the knowledge acquired with the ITIL implementation.

This paper presents the results obtained after a real migration, with the M2CCF, of University Portucalense IT services to a private cloud.

The rest of the paper is organized as follows: Section 2 presents the overview of the cloud computing paradigm, its concerns and governance, the ITIL framework and an overview of various frameworks developed by other authors to support the migration to cloud computing. In Section 3 is presented the framework developed (M2CCF) and a case study of a real implementation of the M2CC. In Section 4 the results of the case study are discusses and analysed. Finally, the paper concludes in Section 5.

2 BACKGROUND

In this section is presented an overview of the CC paradigm as well as ITIL, since, as we argue, it is a companion of the M2CCF in the migration to cloud computing.

Cardoso, A. and Moreira, F.
Migration Results to a Private Cloud by using the M2CCF.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 95-102
ISBN: 978-989-758-187-8

2.1 Cloud Computing

The term "cloud computing" (CC) was coined in the fourth quarter of 2007, in the context of a joint project between IBM and Google (S. Zhang, Zhang, Chen, and Huo, 2010). One definition recognized by several authors, such as, (Foster, et al., 2008; Zhang, et al., 2010), considered as being holistic (Swamy, 2013) and adopting a broad scope is the one presented by The National Institute of Standards and Technology (NIST). According to that definition the CC is classified in four deployment models: public, private, hybrid and community. Each of the aforementioned deployment models is divided into three layers (also known as service models), according to the services it provides to the users (Mell and Grance, 2011; Vaquero, et al., 2009). These layers are, on the first level, Infrastructure as a Service (IaaS), where the user can afford, upon request, processor resources, storage and networking, among others. On a second level, the Platform as a Service (PaaS) layer allows users to implement their applications in the cloud, by using the programming languages and tools provided by the cloud service provider. The third layer corresponds to Software as a Service (SaaS), where the applications, provided by the cloud provider, are made available to the costumers.

The CC paradigm offers various advantages, such as the ability to dynamically adjust the resources according to the needs, a great scalability in resource utilization, a reduced initial investment, an easy access, but also has number of challenges that must be overcome. Note however, that some of these challenges are old but in a new scenario (Jansen, 2011). Among the challenges are issues such as the security (Armbrust et al., 2009), the service availability, the lack of knowledge on where is the information stored, the retrieval of the information (for instance at the end of contract or provider bankruptcy), the lack of legislation (it is mandatory to obtain appropriate legal advice) and the costs (the issues are somehow similar to rent or buy a car).

2.2 ITIL

Enterprise activities increasingly rely on the fundamental support of IT to sustain the growth of the business. Amongst the IT governance frameworks, ITIL gains prominence on the migration to CC because, as stated by (Sahibudin, et al., 2008), implementers should use ITIL to define strategies, plans and processes, which are the key actions to migrate to CC. Furthermore, ITIL is chosen by its acceptance. Indeed ITIL is the most widely adopted approach for IT (Mourad and Johari, 2014), with an acceptance of 28% followed by COBIT with 12,9% (ISACA, 2011).

The ITIL is a de facto standard and the reference model for IT management processes. This model was developed by the English government for use in IT companies, and was quickly adopted across Europe as the standard for best practice in service delivery IT.

Published by the Central Communications and Telecommunications Agency (CCTA) and, more recently, the Office of Government Commerce (OGC), ITIL provides a practical, no-nonsense framework for identifying, planning, delivering and supporting IT services to the business. Consisting of a set of good practices, described over five volumes known as Service Strategy, Service Design, Service Transition, Service Operation and Continual Service Improvement, ITIL is currently in version 3 (known as ITILv3 and ITIL 2011 edition). Its last update was in 2011, ITILv3 it has been rapidly adopted throughout Europe as the de facto standard for best practices in IT service delivery.

2.3 Migration to Cloud Computing, Frameworks

Several authors investigated the migration to CC. Accordingly, in this section, we expose a summary of these works. More details and a comparative study of these works can be found in (Cardoso, Moreira, and Simões, 2014).

Among the works developed for migration to CC is the work of Vivek Kundra (2011) that proposes a decision framework for CC migration. Adela Tušanová (2012) suggest a six step framework. Ali Khajeh-Hosseini et al., in (Khajeh-Hosseini, et al., 2010b), describe the challenges that a decision maker faces when assessing the feasibility of the CC migration in their organizations, and presents the Cloud Adoption Toolkit, which has been developed to support this process.

Ezzat et al. in (2011) proposes a framework focused to support decision makers, in their migration to CC, depending on their own business cases and predefined issues. They view the migration to CC under three perspectives, the business, the technical and the economic ones. In (Chauhan and Babar, 2011) the authors summarize their practical experience by reporting the information gathered when they migrated the Hackystat open-source software's framework, to the CC. Patricia V. Beserra et al., in (Beserra, et al., 2012) present Cloudstep, a step-by-step decision

process aimed at supporting legacy application migration to the CC. The process was exemplified with the migration of a medical commercial application to the CC. The approach followed by Frey et al. (2012), CloudMIG for migration to CC, aims at supporting SaaS providers in the comparison and planning phases to migrate enterprise software systems to IaaS or PaaS based clouds. Banerjee in (2012) addresses the migration to CC of enterprise level workloads without redesigning or re-engineering the existing applications. The Innovation Value Institute (IVI) from the National University of Ireland Maynooth ("Innovation Value Institute (IVI)," n.d.) consortium to address the issues involved in the CC migration developed and tested a life cycle for systematically managing cloud migration projects, the IVI Cloud Computing Life Cycle (Conway and Curry, 2013).

According to the analysed documents, the majority of the studied frameworks do not include an initial step to define a strategy for the migration of services to CC. Besides that, they do not address risk management nor legal issues either, nor analyses the impact of migrating services to CC. Additionally, the contracts management, the vendor lock-in, the testing of the achieved solution, the use of good practices and the continual improvement of the solution are other issues that are not covered by the analysed solutions.

Notwithstanding each of the studied frameworks offer a solution to migrate IT to CC, none of them points a way to enforce that the actions developed to complete each process (that make up the framework) are managed, done appropriately and in an organized way. To solve this, an IT governance framework, such as ITIL or COBIT, could be used as a reference to define each of the framework processes to achieve the best solution for the organization.

3 MIGRATION TO CLOUD COMPUTING

In this section we shortly present our M2CC framework (Cardoso, et al. 2015), and discuss its application on a real migration of IT services to CC scenario.

3.1 Framework

In the outsourcing processes there is always an interaction between IT's service provider and the customer. Accordingly, we have grouped the activities of the M2CCF, into two major groups, the on premise and the off premise, both aggregating the activities that an organization has to perform when migrating services towards the CC. These groups match the key's stakeholders of this process, that is, the customers and the CSPs.

The on premise group embraces the activities that the organization must solve on their own to migrate services to the CC. Accordingly it consists in four steps, "Define a strategy", "Identify and understand", "Define, select analyse and map" and "Migrate and govern". On its side, the off premise group encompasses processes to provide "Information about cloud services" and the cloud services. Before starting the process of migrating to the CC, the organization must at first identify and understand the business and technical issues, which lead to a migration of services and applications to the CC. Among these issues, there are cost savings, agility and scalability offered by the CC. At the "Define a strategy" process, the organization comprehends the CC concept, identifies the reasons why to migrate services to the CC and develop a strategy plan. In the "Identify and understand" process, the customer performs a full assessment of the infrastructure, services, applications and data, to perceive in full detail its IT, to identify what to migrate and to later compare in-house versus CC solutions. After comprehending his IT, the customer is ready to define a migration's plan in the "Select analyse and map" process. Based on the information of the earlier processes, on the migration plan and in the information gathered from the CSPs, he chooses the most appropriate suppliers for the migration.

A sub-process analyses and ponders the whole information to produce the input to the "Map" sub-process mapping out services to their cloud counterparts or creating new ones. Lastly, in the "Migrate and govern" process, the organization migrates the selected services and applications to the CSPs according to the defined migration's plan. The migration is performed with the joint participation of the IT department, business, CSPs and with the service integrator (where appropriated). This migration may be phased, and there must be a validation by the end of each phase, according to the customer needs. Finally, the customer collects information regarding the performance of the CSPs and checks if they are in accordance as specified in the contracts and the SLA.

3.2 Case Study

Taking into account, on the one side, the benefits

that the methodology of the case study expose in investigating real life phenomena (Yin, 2003) and for the other hand the prescription the same author does of it, we consider the case study to be the adequate choice of a methodological approach to comprehend and validate the problem under the research. In the subsequent paragraphs, a case study of migrating services and applications to the CC (developed at the University Portucalense Infante D. Henrique - herein referred to as UPT, a typical higher education institution) is presented. The aim of this case study is to understand, explore and describe the migration's process to the CC under the framework developed in (Cardoso et al., 2015) as well as its relationship with ITIL.

The working methodology towards this case study entailed a close monitoring of all stages leading to the migration of services to the CC, the participation in the decision-making process and the intervention in the whole process of installing a private cloud and on the migration of services to the implemented environment.

All of the work in this case study took place in accordance with the guidelines of the framework developed. Thus, the UPT IT's area starts the whole process by defining the initial strategy with broad outlines and guidelines for the whole process. This phase is followed by the stage of identification and understanding of all services, applications and data in use by the UPT area of IT. This stage is vital to establish a strong foundation of information for pursuing the subsequent phases of the framework.

The next process, "Define, select, analyse and map" begins by defining a migration's plan, which delineates all the details of the migration process itself. Concomitantly, it runs the "Select providers" sub-process where the proposals are analysed and the suppliers that best fit the needs identified are selected. Following these processes and according to the information added so far, the team responsible by the migration's process analyses the aspects of the solution and conducts, in collaboration with the supplier, a test to the solution. The "Map" sub-process follows near the end of this process. It is a sub-process where the services' applications and data, defined to be migrated to the CC, and the corresponding CC services are matched. The migration's process ends with the physical migration of the selected services to the CC's environment. This process also includes a pilot test of the entire solution as well as the training of actors and the beginning of a continuous process of monitoring and improvement of the solution as a whole.

3.3 Results Obtained

In technical terms the implemented solution improved the operating conditions of the services migrated to the private cloud. This improvement was reflected at the level of management, which became centralized and was carried out in a much more automated way and the services provided require now considerably less time in order to produce results.

Based on the research developed and on the UPT IT needs identified it was confirmed that the private cloud is the best suitable solution. The tasks of the M2CC framework were performed according to ITIL. For every process of the M2CC's framework, we found out that ITIL has support, except in what relates to the management of the IT's staff.

4 DISCUSSION

To support the validation of the framework developed, amongst other actions that are not within the scope of this paper, a case study was accomplished.

4.1 Migration Results

The findings of the case study indicate a close relationship between the process identified in the M2CC's framework to migrate IT to the CC and ITIL, although some aspects, like the management of IT's human resources and the project management, are not covered by ITIL. All other are sufficient to cover the demands of the M2CC's framework.

The first ITIL book, "Service Strategy", provides guidance on clarification and prioritization of service provider investments in services. As such, the major usage occurs is in the first two processes of the M2CC's framework – the first defining a strategy and the second gathering information concerning the current state of the in-house IT's infrastructure. The second book, "Service Design", aims to design appropriate and innovative services to meet the business requirements. Hence, its major usage is on the gathering information from the current IT's infrastructure and definition of the services in the cloud. Hereinafter is the "Service Transition" book, and as the name suggests, it takes care of the transition of services, that is, builds and deploys IT services. Its major usage, is in the "Migrate and Govern" M2CC framework's process, which is the process that is responsible for the real

migration of services, applications and data to the CC. This last process of the M2CC's framework, is also responsible for the services in the cloud environment functioning and its improvement. As such, the last of the ITIL books, "Service Operation" and "Continual Service improvement" provides major support to this process.

Despite the M2CC framework widely benefits from ITIL, it is not required that the organization previously implements ITIL so that it is able to perform the migration to the CC. The team responsible for the migration to the cloud may only implement the necessary ITIL processes to gather the required information or to manage some processes. However, it should be pointed out that if the organization already follows the ITIL framework, the usage of the M2CC framework is simplified.

The case study allowed us to validate the framework developed and solve minor issues. For example, the task "Test the solution in a controlled environment" was added because of a practical necessity by the time of the cloud's implementation. According to our study, the use of good practices to implement the migration of services to the CC benefits the organizations, mainly because they can reuse the majority of the work performed by them when deploying the good practices' framework and their results in the migration to the CC. Additionally, when both the customer and the CSP have implemented the same good practices, they have a common language facilitating therefore their communication.

To further, validate the results achieved, three interviews were conducted with the UPT staff that has a direct contact with the services migrated to the private cloud.

4.2 Interviews

The first interview, done to 16 employees, is intended to validate the tests performed in a controlled environment to validate the details of the achieved solution. The interviews questions and the results are depicted on Figure 1

According to the results, the tests provide the expected outcomes - the migration to the CC of the selected services and applications was successful and the technical difficulties were overcome.

Interview II was accomplished to evaluate the perception of the staff directly affected by the migration to the CC (a total of 16 interviews). Taking into account the answers to questions, one, two and three, of the conducted survey, see Figure 2, we can conclude that 100% of the users consider that the functionalities of the applications remains unchanged and not suffer any access breaks during the migration's period.

In terms of access, the users also recognize that there had been no change in accessing the applications they already had. The previous and detailed analysis of the applications used and the users' habits led to selecting a period for the migration of applications that would have less impact to the users.

This analysis was performed in the process of "Identify and Understand" and reinforced in the "Define, select, analyse and Map" particularly in "Define a Migration Plan" for example in the access definition.

Questions four and five refer to the troubleshooting, equated in the initial phase of implementation of the framework, in the process "Define a strategy" when defining "Why move to cloud". It is widely spread view that the speed of access to services has been improved and the availability problems were solved. Some of these, still felt by some users, are due to intermediate servers that have not yet migrated to the new solution, such as proxy servers.

Question six aims to assess the initial study of the information that each application uses. According to the feedback from the users, all the information used by applications has been preserved

1. Was the outcome of the tests performed the expected?
2. Has the access speed to the applications has been improved?
3. Were there breaks in the applications availability?
4. Were the accesses to the applications kept unchanged?

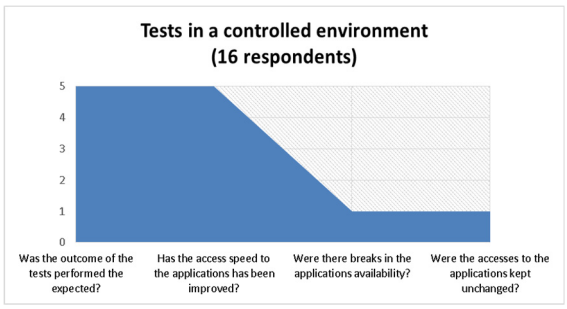


Figure 1: Interview I.

1. Were the applications were migrated with the same functionalities?
2. Were there were breaks in the application accesses at migration time?
3. Were the accesses to the applications kept unchanged?
4. Has the access speed to the applications has been improved?
5. Have the applications' availability problems have been solved?
6. Was the information migrated to the new platform was complete?
7. Did your processes, dependent on migrated applications, suffer any changes?
8. Were the applications chosen to be migrated to the cloud were appropriate?
9. Was it important to maintain the existing applications?
10. Was it important to maintain the applications on the premises of UPT?

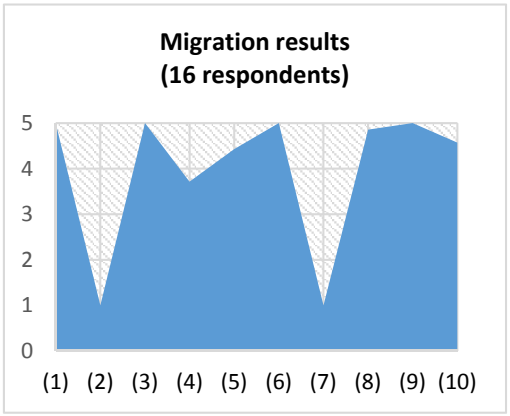


Figure 2: Interview II.

The implemented solution allowed:

1. To improve the backup process?
2. To dynamically change the resources allocated to machines?
3. To increase the service's availability?
4. To improve the access speed to the applications?
5. To help you create new machines?
6. Decreasing the amount of time needed to create a new machine and providingnew services?
7. To reduce the number of physical machines?
8. To reduce the consumption of electrical energy?
9. To turn on and off machines according to the needs?
10. To create machines for testing and delete them when they are no longer needed?
11. To generated new machines from a template?

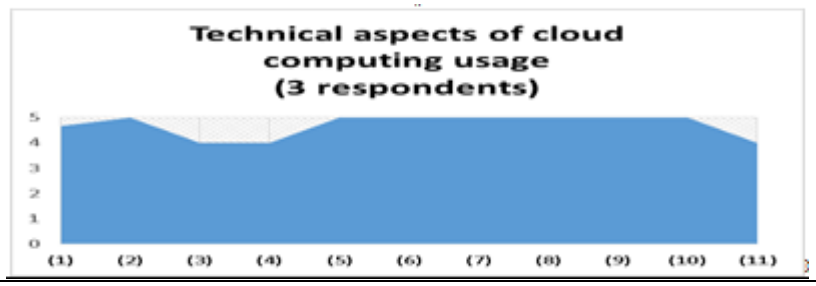


Figure 3: Interview III.

in the new environment. This information was collected in connection with the "Identify and Understand".

According to the answers obtained to questions seven, eight and nine, we found out that the migration did not cause any disruptions during normal day-to-day of the users. These issues resulted from the validation of processes "Identify and Understand", "Analyse and test" and "Migrate and govern", for example in gathering information about services in the process "Identify and Understand", in analysing the impact on the sub-process "Analyse and test" and in selecting applications to migrate from sub-process "Define a migration plan", namely "What to move".

Question 10 relates to the cloud type chosen in the sub-process "Defines the migration plan". However, it also comprehends the security required by the institution validated in the sub-process "Define a migration plan", with the risks discussed in the sub-process "Analyse and test" and the sub-process "Select the providers".

Interview III, see Figure 3, aims to validate the fact that the solution achieved encompasses the CC's technical advantages. This interview has showed that the solution achieved includes various CC advantages such as, the capacity to dynamically change the resources of the machines, facilitate the

task of creating new machines, activate and deactivate machines according to the needs and the possibility of having templates to create new machines based on a common configuration. Moreover, the solution also solves some problems found, such as improving the backup processes, increase the service's availability and the speed to access these services.

5 CONCLUSIONS

IT managers are increasingly concerned in minimizing investments; capitalize on investments already made and the way the services are performed to achieve greater productivity with lesser costs. The CC is a paradigm that allows customers to start new services or expand already existent ones without requiring large upfront investments, enabling customers to acquire and release resources dynamically according to their needs in a pay-as-you-go form. One of the main challenges facing the migration to this new paradigm is the need to review and to adapt the services and IT processes to operate in the new paradigm. Another issue arises from the difficulty of bringing services back to the environment they had before, after they have migrated to the cloud. One other issue occurs from the costs involved in the migration. Therefore, the migration to CC must be carefully planned and performed. So, it is important to investigate how the organizations can efficiently and effectively migrate IT from the conventional model to CC.

Taking into account, the need to better meet the user requirements with lower costs, the advantages of the CC, the advantages of ITIL in managing IT services (with its major acceptance and adoption's index compared with other service management frameworks) and the possibility to use the information gathered by ITIL, the work developed examined the adequacy of ITIL in the migration of traditional IT environments to CC. By creating a framework to migrate services to CC and mapping the processes of the framework to the ITIL processes, we validate the applicability of ITIL to the migration to CC.

Bearing in mind that there are some interdependencies among the ITIL processes and that implementing the whole ITIL is not an easy task, we purpose, as a future work, to develop a "mini-ITIL" to support the Small and Medium Enterprises (SMEs) that have not implemented ITIL, in the migration to the CC.

REFERENCES

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R. H., Konwinski, A., Lee, G., et al. 2009. Above the Clouds: A Berkeley View of Cloud Computing (No. UCB/EECS-2009-28). EECS Department, University of California, Berkeley. http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.html.

Banerjee, J. 2012. Moving to the cloud: Workload migration techniques and approaches. *19th International Conference on High Performance Computing*, 1–6.

Beserra, P. V., Camara, A., Ximenes, R., Albuquerque, A. B., and Mendonca, N. C. 2012. Cloudstep: A step-by-step decision process to support legacy application migration to the cloud. *IEEE 6th International Workshop on the Maintenance and Evolution of Service-Oriented and Cloud-Based Systems,* 7–16. doi:http://dx.doi.org/10.1109/MESOCA.2012.6392602.

Cardoso, A., Moreira, F., and Simões, P. 2014. A Survey of Cloud Computing Migration Issues and Frameworks. In Á. Rocha, A. M. Correia, F. B. Tan, and K. A. Stroetmann (Eds.), *Advances in Intelligent Systems and Computing: New Perspectives in Information Systems and Technologies*, Vol. 275, 161–170. Springer International Publishing.

Cardoso, A., Moreira, F., and Simões, P. 2015. A Support Framework for Migration of e-Government Services to the Cloud. In *P. Z. Mahmood (Ed.), Cloud Computing Technologies for Connected Government*. IGI Global Publication.

Chauhan, M. A., and Babar, M. A. 2011. Migrating Service-Oriented System to Cloud Computing: An Experience Report. *2011 IEEE International Conference on Cloud Computing (CLOUD)*, 404–411.

Conway, G., Curry, E. 2013. The IVI Cloud Computing Life Cycle. *Cloud Computing and Services Science. Springer.* http://www.edwardcurry.org/publications/Conway_IVILifecycle.pdf.

DMTF to Develop Standards for Managing a Cloud Computing Environment. (n.d.). http://www.dmtf.org/about/cloud-incubator/DMTF_Cloud_Incubator_PR_FI N.pdf.

Ezzat, E. M., Zanfaly, D. S. E., Kota, M. M. 2011. Fly over clouds or drive through the crowd: A cloud adoption framework. *International Conference and Workshop on Current Trends in Information Technology*, 6–11.

Foster, I., Zhao, Y., Raicu, I., Lu, S. 2008. Cloud Computing and Grid Computing 360-Degree Compared. *IEEE. Grid Computing Environments Workshop,* 1–10.

Frey, S., Hasselbring, W., Schnoor, B. 2012. Automatic conformance checking for migrating software systems to cloud infrastructures and platforms. *Journal of Software: Evolution and Process*, 1.

Heininger, R. 2012. IT Service Management in a Cloud Environment: A Literature Review. *Proceedings of 9th Workshop on Information Systems and Services Sciences.*

Innovation Value Institute (IVI). (n.d.). http://ivi.nuim.ie/

ISACA, I. 2011. Global status report on the governance of enterprise IT. http://www.isaca.org/Knowledge-Center/Research/Documents/Global-Status-Report-GEIT-10Jan2011-Research.pdf.

Jansen, W. A. 2011. Cloud Hooks: Security and Privacy Issues *44th Hawaii International Conference in Cloud Computing. System Sciences.* 1–10.

Khajeh-Hosseini, A., Greenwood, D., Smith, J. W., Sommerville, I. 2010b. The Cloud Adoption Toolkit: Supporting Cloud Adoption Decisions in the Enterprise. *Software: Practice and Experience*, abs/1008.1900(4), 447–465.

Khajeh-Hosseini, A., Sommerville, I., Bogaerts, J., Teregowda, P. 2011. Decision Support Tools for Cloud Migration in the Enterprise. *IEEE International Conference on Cloud Computing*. 541–548. doi:http://dx.doi.org/10.1109/CLOUD.2011.59.

Kumar, V., Garg, K. K. 2012. Migration of Services to the Cloud Environment: Challenges and Best Practices. *International Journal of Computer Applications*, 55(1), 1–6.

Kundra, V. 2011. Federal cloud computing strategy. Office of the CIO, Whitehouse. *White House,[Chief Information Officers Council.* http://www.cio.gov/documents/Federal-Cloud-Computing-Strategy.pdf.

Mell, P., Grance, T. 2011. The NIST Definition of Cloud Computing. *NIST*. http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf.

Mourad, M. B. A., Johari, R. 2014. Resolution of Challenges That Are Facing Organizations before ITIL Implementation. *International Journal of Future Computer and Communication*, 3(3), 210–215.

Nkhoma, M., Dang, D. 2013. Contributing factors of cloud computing adoption: a technology-organisation-environment framework approach. *International Journal of Information Systems and Engineering*, 1(1), 38–49.

Sahibudin, S., Sharifi, M., Ayat, M. 2008. Combining ITIL, COBIT and ISO/IEC 27002 in Order to Design a Comprehensive IT Framework in Organizations. *Second Asia International Conference on Modeling Simulation.* 749–753.

Swamy, S. 2013. Cloud Computing Adoption Journey within Organizations. *In P. L. P. Rau (Ed.), Cross-Cultural Design. Cultural Differences in Everyday Life, Lecture Notes in Computer Science,* Vol. 8024, 70–78. Springer.

Tušanová, A. 2012. Decision-making framework for adoption of cloud computing. In Cs. I. D. D. M. N. Prof. Ing. Alena Pietriková (Ed.), *Proceedings from the 12th Scientific Conference of Young Researchers*. http://web.tuke.sk/scyr/data/templates/Proceedings_2012.pdf.

Vaquero, L. M., Rodero-Merino, L., Caceres, J., Lindner, M. 2009. A break in the clouds: towards a cloud definition. S*IGCOMM Comput. Commun. Rev.*, 39(1), 50–55.

Yin, R. K. 2003. Case Study Research: Design and Methods. *Applied Social Research Methods. SAGE Publications*. http://books.google.com.my/books?id=BWea_9ZGQMwC.

Zhang, Q., Cheng, L., Boutaba, R. 2010. Cloud computing: state-of-the-art and research challenges. *Journal of Internet Services and Applications*, 1(1), 7–18.

Zhang, S., Zhang, S., Chen, X., Huo, X. 2010. Cloud Computing Research and Development Trend. *Second International Conference on Future Networks.* 93–97.

Assessment of Factors Influencing Business Process Harmonization
A Case Study in an Industrial Company

J. J. M. Trienekens[1], H. L. Romero[2] and L. Cuenca[3]
[1]*Department of MST, Open University, Heerlen, The Netherlands*
[2]*Department of TFDO, ASML, Eindhoven, The Netherlands*
[3]*Research Centre on Production Management and Engineering (CIGIP), University of Polytechnics Valencia, Valencia, Spain*
jos.trienekens@ou.nl, llcuenca@omp.upv.es

Keywords: Case Study, Contextual Factors, Process Harmonization, IT Integration.

Abstract: While process harmonization is increasingly mentioned and unanimously associated with several benefits, there is a need for more understanding of how it contributes to business process redesign and improvement. This paper presents the application, in an industrial case study, of a conceptual harmonization model on the relationship between drivers and effects of process harmonization. The drivers are called contextual factors which influence harmonization. Assessment of these contextual factors in a particular business domain, clarifies the extent of harmonization that can be achieved, or that should be strived at. From both qualitative, as well as some quantitative, assessment results, insights are being discussed on the extent of harmonization that can be achieved, and on action plans regarding business (process) harmonization and (IT) integration.

1 INTRODUCTION

The interest in process harmonization by researchers and practitioners has increased in the last years (Fernandez and Bhat, 2010), (Romero, 2014). The process of harmonization is considered as the elimination of differences and inconsistencies among processes in order to make them uniform or mutually compatible (Pardo et al, 2012). Harmonization of processes will lead to effective robust business processes (Siviy et al, 2008), cycle-time reduction and overall operational efficiency (Kumar and Harms, 2004). With process harmonization different business process domains can be integrated, their efficiency and performance can be improved. E.g. the reduction in the number of process variants decreases the costs of process maintenance and increases the agility towards process changes (Manrodt and Vitasek, 2004). However, recognizing similarities and differences between processes and identifying harmonization opportunities is difficult, in particular when dealing with processes in a multi-model environment. Therefore, a trade-off has to be distinguished between the costs and the benefits of striving at totally harmonized business process domains or allowing the business domains, and their processes, to have local relevant variations (Tregear, 2010). In (Romero, 2014) a conceptual harmonization model is presented on the relationship between drivers and effects of process harmonization. These drivers are called contextual factors. Assessment in a particular business domain, clarifies the extent of harmonization of business processes that can be achieved. This paper presents the application of the mentioned harmonization model in a case study in industrial practice, i.e. at DEKRA, an international certification body in The Netherlands. DEKRA is confronted with challenges regarding performance problems and inefficiencies in their testing and certification services. In conformance with their international business strategy, standardization, integration, and improvement are key strategic terms on the higher management levels of the multi-national company. Currently, one of the main questions is to what extent business process domains can, or should be, harmonized. Improvement projects in the recent past have shown that company-wide or even process domain-wide, improvement projects are time-consuming and limited regarding their effectiveness. The objective of the case study is three-fold: first, to investigate whether the conceptual harmonization model can be made operational in an industrial case study; second, to assess the contextual factors, which influence the extent of harmonization that can be achieved in the particular situation; and third, to

Trienekens, J., Romero, H. and Cuenca, L.
Assessment of Factors Influencing Business Process Harmonization - A Case Study in an Industrial Company.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 103-110
ISBN: 978-989-758-187-8

derive so-called focus areas for business (process) improvement. The structure of this paper is as follows. Section 2 introduces the conceptual harmonization model. In Section 3 the case study characteristics will be addressed. Section 4 presents the application and the validation of the conceptual harmonization model. Sections 5 and 6, i.e. lessons learned and discussion, finalize the paper.

2 THE CONCEPTUAL MODEL

The first part of the model, see Figure 1, distinguishes three different levels in the organizational context: external, internal and immediate. Each level includes a set of contextual factors. The second part presents six aspects of process harmonization which can be differentiated when evaluating the level of harmonization of business processes. These aspects have been derived from a set of indicators, as described in literature to measure the level of harmonization. Their interrelations with the contextual factors have been empirically investigated in case studies. The conceptual model suggests that when analyzing the effect of contextual factors, one should not only consider harmonization of a process as a whole, but also consider harmonization of particular aspects of a process. The third part concerns the elements of business performance that are affected by changes in the level of process harmonization, but this part is out of scope of this paper. See for more details on this part (Romero, 2014).

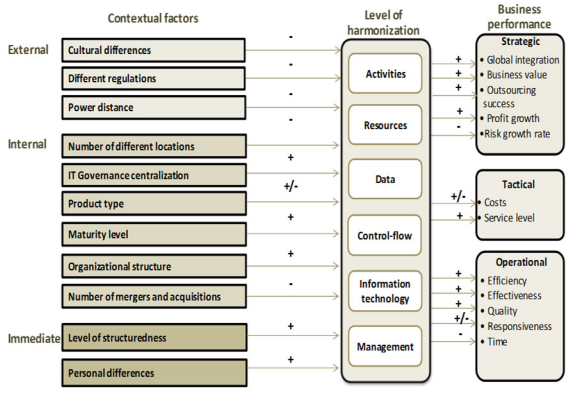


Figure 1: A conceptual model on process harmonization.

In this paper, our focus is on the first and second part of the model, which concerns the effect of contextual factors on different aspects of process harmonization. The external factors characterize the business network in which the organization operates and that are beyond the control of an individual organization. Three external factors were identified: cultural differences, different regulations and power distance. (Ang and Massingham, 2007) discussed the greater the 'cultural differences', the greater the difficulty in knowledge transfer across cultures. There are mandatory and unavoidable variations that come from 'differences in regulations' such as financial regulations, taxation regimes, import/export regulations and employment practices (Tregear, 2010). 'Power distance' refers to differences in the relationship among firms in inter-firm collaborations. Organizations with low power distance have a higher level of integration, i.e. harmonization, of their business practices, while those with medium and high power distance had a low level of integration. Internal factors describe the internal environment of an organization. Seven internal factors are included in our model. It is expected in literature that a higher 'number of locations' decreases the level of process harmonization. However, the effect of multiple locations is not straightforward because it is mixed with other factors, such as 'legal requirements', 'personal differences' among individuals performing the same tasks in different locations and 'cultural differences' (Tregear, 2010). The second internal factor is 'IT governance centralization'. This factor leads to a higher level of harmonization, but in some cases the initial investment needed to centralize, e.g. IT infrastructure, is too high, and savings that can be achieved through process harmonization do not balance this investment (Buchta et al, 2007). 'Product type': more differences in products and services may require variation in the processes that create, deliver and maintain them (Tregear, 2010), suggesting a decrease in the expected level of process harmonization. 'Maturity level': it has been observed that organizations which perform better in their harmonization initiatives, have at least a moderate level of process maturity (Rosenkranz et al, 2010). 'Organizational structure' was also identified as an internal factor that exerts an influence in the level of harmonization (Girod and Bellin, 2011). Regarding the factor 'organizational structure', two dimensions have been identified from literature, respectively centralization and formalization (Romero, 2014). The aspects of centralization include: personal participation in decision making, hierarchy of authority, and departmental participation in decision making. Regarding organizational structure formalization four aspects are identified, including: job codification, job specificity, rule observation and written communication. The last internal factor is the number of 'mergers and acquisitions' that have taken

place in the organization. This factor definitely decreases the level of harmonization of business processes, by increasing the number of process variants that coexists. The harmonization of these variants consolidates processing volumes and allows the organization to exploit economies of scale. Finally, the immediate factors define the process under study, including: level of structuredness and personal differences. "Non routine processes are less applicable to harmonization than routine processes" (Rosenkranz et al, 2010). An argument to support this statement is that different parts of a process need to be open for creative decision making, while others have to meet legal requirements of different countries. There are also unstructured, unmeasured and unrepeatable processes that can lead to a low level of harmonization (Lillrank, 2003). The potential of a process to be successfully harmonized also depends on personal differences such as level of experience and knowledge of the people involved in the process. The lack of interpretative assessment via employees during a process suggests that harmonizing this process is possible and leads to a successful harmonization process. Regarding the six main harmonization aspects of business processes: activities refer to the level of harmonization of specific steps in the process. Control-flow measures the level of harmonization of the sequence of activities. Data measures the level of harmonization of input and output data used in the process. Information technology refers to the level of harmonization of IT systems. Management measures the harmonization of the process assessment, and resources refer to the level of harmonization of human resources involved in the process (Tregear, 2010).

3 CASE STUDY

3.1 Case Study Characteristics

Within the DEKRA business unit of Industrial Services, DEKRA Certification is active in three business domains, respectively Product Testing & Certification (e.g. certification of medical devices, consumer goods), Systems Certification (e.g. work safety, environmental and quality management systems), and Certification of Persons (which focuses on the independent testing and certification of technical and management staff in various business areas. The scope of the case study is on the business process domains of Systems Certification (SC), and the Certification of Persons (CP). The main business process within the SC domain includes order preparation, planning, auditing, corrective ac-tions and invoicing. These are carried out for the following SC services, respectively: (initial) certification, surveillance, recertification and decertification. The SC services result e.g. in certification of management systems (ISO9001 certificates), certification of quality management in hospitals (Dutch HKZ certificates) and certification of guidelines for construction companies (BRL certificates). CP has on a high abstraction level, i.e. the main business process, similar activities as SC, but is oriented on different types of services. In the CP domain the subjects to be tested and certified are not quality management systems, but technical and management staff in various technical business domains. These technical experts need to be assessed periodically, with respect to their skills and knowledge on work safety, e.g. in the energy supply domain. The basis for certification in the CP domain is the independent, reliable and fair examination of persons. The examinations are based on so-called certification schemes which reflect (international) criteria for certification. Although DEKRA Certification is responsible for the examination, these processes are being outsourced to independent examination institutes. In the CP domain the services include on the one hand the quality insurance of the examinations, the knowledge, expertise and behavior of examiners, the examination locations, and on the other hand the analysis of examination results, and the certification of persons. Over the last years the number of certified persons in the CP domain has increased rapidly, largely due to the energy market where safety is becoming an issue of increasing importance. Various certification processes for persons have been developed under time pressure and often independently from each other. Part of the CP processes is currently separately managed by different product experts and certification managers. Although there are similarities between the SC and the CP process domains, and also within both domains, the differences are increasing. In order to investigate directions for improvement, it was decided to assess the extent of harmonization that can be achieved, or that should be strived at, in the particular business domains.

3.2 Case Study Methodology

The case study distinguishes a preparation phase, a data collection phase and a data analysis phase. In the preparation phase the case study scope has been determined (i.e. SC and CP), the processes and their

differences and similarities have been investigated and the information sources have been defined. The objective of the preparation is to understand the existing processes. Due to the fact that the process descriptions of SC and CP differ, it was needed to model the current processes into a similar (BPMN) format to get a consistent reference frame for the interviews. Regarding data collection two types of information sources have been used: documents and interviews. The documentation included business presentations and process descriptions of the SC and CP business processes. Interviews have the benefit to specifically focus on the case study topic. The interviews were conducted using a pre-defined questionnaire (Romero, 2014). The questionnaire consists of three parts, respectively questions on organizational characteristics, questions to assess the contextual factors under study and questions to assess the process structuredness. The first part is on characteristics such as company type, size and age. This information can be used for comparison of data from previous or subsequent case studies. The second part of the interview assesses the contextual factors. Appendix 1 presents as an example the specific close-ended questions regarding the organizational structure (centralization and formalization) that were used to assess this factor and the Likert scales used, (Romero, 2014). The Appendix shows also the calculation of the assessment results of the factor organizational structure. The close-ended questions facilitate the comparability of data and of the data with previous literature. Also combined questions have been used, i.e. questions that start with a close-ended part and based on the choice made, additional explanations in an open-ended format are asked. The use of semi-structured interviews in this case study is motivated by the fact that this type of inquiry is exploratory and the interviews should allow for unexpected information, e.g. on assessment factor interpretations of the interviewees. While choosing the right type of interview is crucial, also the selection of the proper interviewees is critical. Considering the case study's research questions, we selected as key informants respectively experienced managers of the certification processes (i.e. product/service managers) and experienced operational certification experts (i.e. lead auditors). In the data analysis phase the assessment results have been analyzed and subsequently discussed and validated in workshop sessions with the four interviewees. In this phase also propositions, from previous case studies (Romero, 2014), on the interrelations between contextual factors and harmonization aspects have been used to derive conclusions. Subsequently per contextual factor, improvement actions have been defined and presented to the general management at DEKRA Certification. In the next section the case study results are presented and discussed.

4 THE CASE STUDY RESULTS

First we will reflect on the business process investigation which preceded the assessment of the contextual factors influencing harmonization. Then we will present the results of the application of the harmonization model. Subsequently we will address how actions for business process improvement have been defined.

4.1 Investigation of the Business Processes

The SC and CP processes have been analyzed, e.g. regarding the modeling languages used, the types of documents and their formats. The similarities and differences, have been discussed and validated with experienced managers in the particular DEKRA business domain. In the context of this case study we point to the following findings from the investigation. In the SC domain, certification services are carried out that are slightly different from each other, i.e. the different types of quality management system certifications (e.g. ISO, HKZ). However, the CP domain has emerged over the last five years with many new certification services which show many differences, both in structure and language. Although there exist on a high level of abstraction one main process model for certification (with defined activities), the SC and CP processes differ with respect to modeling language used, levels of detail in process elaborations, and document formats (i.e. work instructions, procedures). SC and CP also have different monitoring and control units. In SC monitoring and control is highly centralized (in a so-called Project Office). In CP this is different with many decentralized control units for the distinct certification services. Regarding the monitoring and control in the SC domain a 'Plan board' application is being used. However, this application system does not support the processes of the CP domain. Further, only the SC processes are modeled and visualized by process flows in the Quality Management System (QMS), an information system that serves as a support for the various certification experts (auditors, product experts and certification managers). The CP domain lacks visualized processes and the QMS only

contains CP standard document formats, procedures and work instructions. These findings from the process investigations were considered as a useful process reference framework for the execution of the semi-structured interviews.

4.2 Assessment of the Factors

Four semi-structured interviews have been carried out, i.e. two interviews with product/service managers and two with lead auditors. In the following we will, for each of the contextual factors, present and discuss the results of the assessment. To illustrate the analysis and the way we came to our conclusions, we will refer in the following for one of the internal contextual factors, i.e. organizational structure, in more detail to the collection of data and the calculation of the results from the close-ended interview questions.

Regarding the external contextual factors, 'Cultural differences' are considered as a factor which is of importance in case the scope of harmonization covers more countries or regions. However, this case study focuses at the particular SC and CP domains at DEKRA Certification. Both domains are monitored and controlled from one central management level at DEKRA Certification. Knowledge transfer on systems certification and certification of persons mainly takes place within the company in The Netherlands. As a consequence the contextual factor 'Cultural differences' does not influence the extent of harmonization that can be achieved. Regarding the factor 'Different regulations', the DEKRA domains SC and CP should meet different types of standards and requirements, e.g. as specified by the Dutch Council for Accreditation. For example, the processes of CP should meet the requirements defined in ISO/IEC 17024:2012, such as the security of examination data and the independability of examination processes. SC should meet other ISO/IEC standards, such as ISO9001 with respect to the quality monitoring and control of business processes and management systems. As a consequence the SC and CP processes show differences, both between and within the domains, and there is a danger of ending up with multiple variations of both SC and CP certification processes. Therefore, the factor 'Different regulations' influences negatively the extent of harmonization that can be achieved.

Regarding 'Power distance' both the SC and CP process domains are, at the highest management level, being monitored and controlled by the same management team. However with respect to the management of the SC certification processes the differences in customer relations cause differences in planning and control. In the SC domain particular customer types are allowed, to some extent, to determine the planning and the scheduling of the certification projects. Auditing and certification, in particular the timing aspects, are here to a large extent tailored to the needs and the wishes of the customer. However in the CP domain, auditing and certification processes are planned and scheduled only by the management team. These kinds of differences in 'Power distance' influence negatively the extent of harmonization that can be achieved at DEKRA Certification.

Regarding the internal contextual factors, both the SC and CP process domain are located at the same industrial area in The Netherlands. So, the factor 'Number of different locations' is 'low'. It also appeared that both domains are able to exchange auditors for particular types of auditing projects. Regarding the factor 'IT governance centralization' it became clear that although decision making regarding IT alignment at DEKRA is formally centralized, the IT landscape shows a rather scattered picture. The SC and CP process domains are partly supported by different systems, even in similar functional areas. This causes that, although IT-governance is formally centralized, there is a negative influence, from the scattered IT-landscape, on the extent of harmonization that can be achieved. Regarding the factor 'Product type', the domains SC and CP have different products (i.e. services) and customers in different market segments. E.g. certification of business systems only makes use of a restricted set of certification schemes, while for the certification of persons many (i.e. >50) certification schemes are being used. Also product/service innovation has different characteristics in both the SC (e.g. long-term, generic) and CP domain (e.g. mid-term, specific). It was concluded that different roles in both the SC and the CP domain are not yet sufficiently defined and implemented. As a consequence the factor 'Product type' influences negatively the extent of harmonization that can be achieved. The 'Maturity level', led to different scores for the SC and the CP domain. In the SC domain a process maturity level 3 was reached, e.g. based on the formal and stable system certification procedures in this domain. However in the CP domain, the process maturity reached is between level 1 and 2. This is caused by the fast growth of the domain over the last five years, and the large diversity of new certification schemes developed. As a consequence it was concluded that the restricted 'Maturity levels'

influence negatively the extent of harmonization that can be achieved. Regarding the factor 'Organizational structure', and its two dimensions centralization and formalization, the Appendix reflects some detailed assessment results to illustrate the close-ended questions as well as the calculation of the scores. The factor has been assessed on the basis of 11 sub-questions on centralization and 14 sub-questions on formalization. The 5-point Likert scale scores and the 4-point Likert scale scores, both derived from literature, are normalized in the Appendix. The score for centralization is 0.47, which is moderate. It could be concluded that there is an average hierarchical network that does not influence negatively the extent of harmonization. Looking at the score for formalization, i.e. 0.59, it was concluded that formalization could be classified as above average. In total, based on the assessment results of both centralization and formalization, the influence of Organizational structure centralization on the extent of harmonization that can be achieved was concluded to be positive. Regarding the factor 'Number of mergers and acquisitions' DEKRA Certification can be considered as a company that has a restricted activity in this type of managerial practices. Over the last five years, only one small and medium sized enterprise has been acquired and merged. This indicates that the number of new and different process variants, and IT systems, that had to be integrated or implemented is limited. Based on the IT governance centralization analysis in the foregoing, it was concluded that the factor 'Number of mergers and acquisitions' doesn't influence the extent of harmonization that can be achieved.

Regarding the immediate contextual factors, the 'Level of structuredness' is based on process aspects such as the repeatability of processes and creativity needed in decision making. In particular the SC domain the processes are, to an above average level, standardized. The DEKRA main process model acts for SC as a generic model from which specific repetitive processes can be derived. The Level of structuredness in the SC domain influences positively the harmonization of processes. However in the CP domain the situation is different. The various domains of certification, the variety of certification schemes, and the fast increase of certification schemes over the last years has led to a rather low 'Level of structuredness'. Because of the quite large differences in the two process domains SC and CP it was concluded that overall the factor negatively influences the extent of harmonization. Regarding 'Personal differences', DEKRA can be considered as a company with differences in audit and certification experiences and knowledge. In particular in the CP domain a particular knowledge regarding the examination and certification of persons is required. Also the quality assurance of automated examination systems requires a specific expertise and qualification of the auditors. In the SC domain the required expertise and knowledge is oriented at quality management and business systems. These 'Personal Differences' lead to the conclusion that this factor negatively influences the extent of harmonization. Table 1 gives a summary of the foregoing discussion.

Table 1: Influences of factors on harmonization.

Contextual factor	Influence
Cultural differences	None
Different regulations	Negative
Power distance	Negative
Number of locations	None
ITG centralization	Negative
Product type	Negative
Maturity level	Negative
Organizational structure	Positive
# Mergers and acquisitions	None
Level of structuredness	Negative
Personal differences	Negative

4.3 Assessment

In this section we will summarize the discussed assessment results for each of the factor categories and we will present briefly the harmonization actions that have been defined at DEKRA Certification, both on processes as a whole, as on particular aspects (i.e. Activities, Resources, Data, Control-flow, Information technology and Management). From the interviews it appeared that the external contextual factor 'Different regulations' has a negative effect on the extent of harmonization that can be achieved. For DEKRA Certification these standards have to be taken as given, and they cannot be adapted or tailored by a certification body. In particular the periodically upgrade of standards by International Standardization Organizations requires extra effort from certification bodies to stay compliant. The assessment results lead to decisions for two harmonization aspects, respectively: Resources and Management. It was decided to define a harmonization project in that resources, i.e. lead auditors from both the SC and the CP domain will start collaboration on the interpretation, the implementation and the maintenance of the various international certification stand-ards. Further, it was decided that in particular the management of knowledge sharing and standardization of certification activities would be implemented to strive towards a more harmonized

business situation. From the assessment results on the internal contextual factors, the factors 'Maturity level' and 'IT Governance centralization' show clear negative influences. Harmonization actions defined pointed to the harmonization aspects of respectively 'Information technology' and 'Management'. A project has been defined on the integration and standardization of the various IT applications in the SC and CP domain, as well as the monitoring and control (i.e. management) on the IT Governance level. The internal contextual factor 'Organizational structure', respectively centralization and formalization, shows a positive influence regarding the extent of harmonization that can be achieved. However, the internal contextual factor 'Product type' influences harmonization negatively and leads to warnings regarding a too high ambition level. Harmonization actions defined point to the harmonization aspects 'Activities' and 'Resources'. Based on this a joint-project has been defined in the SC and CP domain, to identify criteria for the adoption and development of new services, i.e. the implementation of new certification schemes. The project should lead to a limitation in the variety of the certification activities and the skills and knowledge that is needed, as well as an improvement of the coherence in the resources and the certification activities.

The immediate contextual factors 'Levels of structuredness' and 'Personal differences', internally and directly related to the SC and CP processes under study, show both negative influences on the extent of harmonization that can be achieved. The SC and the CP process domains have independently been managed and have different growth curves with respect to standardization of processes. In particular in the CP domain the fast business growth and increase in certification schemes, has led to an unstructured variety in processes, procedures and work-instructions. Harmonization actions defined, pointed clearly to the harmonization aspects 'Activities', 'Control-flow' and 'Data'. Consequently a project has been defined to cover these aspects. First, the development of so-called Project Office activities in the CP process domain has been defined. Project Office activities bundle the expertise and streamlines the planning and control-flows of the certification services. These process improvements will reduce the throughput time, as auditors are supported by a Project Office and can focus on their job. Next to these new activities in the process models, the procedures and the documents of CP are also rewritten into the same data format as for SC. Advantages include a higher consistency in the quality of the process out-comes, e.g. customer reports.

5 LESSONS LEARNED

The conceptual model on contextual factors and harmonization aspects can be made operational in an industrial environment, and can be used as a valuable assessment tool. An operationalization of this conceptual model led to interesting assessment results. The influences of the factors on harmonization became clear, in particular in the discussions with the selected involved practitioners. Consequently, agreements with respect to different types of improvement actions, in terms of concrete projects on different harmonization aspects (e.g. resources, activities, etc.), could be defined. Thus, the application of the conceptual harmonization model has resulted in consensus in the company on a concrete action plan, e.g. with respect to information systems integration and has proven its value in the particular business situation. Further, the understanding of the factors influencing harmonization, could be used as a valuable input for trade-off decisions.

6 CONCLUSIONS

A single qualitative case study methodology was adopted in this study to identify relations between contextual factors and the level of harmonization. Although interesting and important, the emergent findings are idiosyncratic or related to this single case study. To strengthen the propositions, it is strongly recommended to apply a multiple case study methodology. Such a methodology enables comparisons between, preferably more quantitative, case study results and can identify consistencies in factor-harmonization aspect relationships. Although the paper points to IT project development the implementation of systems are out of scope of this paper, since these will only become available on the mid- and long-term. However, the implementation actions have already become part of the DEKRA business development plan and will be evaluated periodically.

REFERENCES

Ang, Z., and Massingham, P., 2007. National culture and the standardization versus adaptation of knowledge

management. *Journal of Knowledge Management*, 11(2), pp.5-21.

Buchta, D., Eul, M., and Schulte, H., 2007. IT Optimization—Reducing Costs without Diminishing Returns. *Strategic IT Management: Increase value, control performance, reduce costs.* Gabler Verlag 2007, 133-153.

Fernandez, J., and Bhat, J., 2010. *Addressing the Complexities of Global Process Harmonization*. Handbook of Research on Complex Dynamic Process Management: Techniques for Adaptability in Turbulent Environments, IGI Global, pp.368-385.

Girod, S. J., and Bellin, J. B., 2011. Revisiting the "Modern" Multinational Enterprise Theory: An Emerging-market Multinational Perspective. *Research in Global Strategic Management,* 15, 167-210.

Kumar, S., and Harms, R., 2004. Improving business processes for increased operational efficiency: a case study. *Journal of Manufacturing Technology Management,* 15(7), pp.662-674.

Lillrank, P., 2003. The quality of standard, routine and non routine processes. *Organization Studies*, 24(2), 215-233.

Manrodt, K. B., and Vitasek, K. (2004). Global process standardization: a case study. *Journal of Business Logistics,* 25(1), pp.1-23.

Pardo, C., Pino, F. J., Garcia, F., & Piattini, M., 2012. *Identifying Methods and Techniques for the Harmonization of Multiple Process Reference Models*. Dyna,79 (172), pp.85-93.

Romero, H., 2014. *The Role of Contextual Factors in Process Harmonization, Ph.D Thesis, nr. D179*, BETA Research School for Operations Management and Logistics, University of Technology Eindhoven, The Netherlands, 2014.

Rosenkranz, C., Seidel, S., Mendling, J., Schaefermeyer, M., and Recker, J., 2010. Towards a framework for business process standardization. *In Business process management workshops (pp. 53-63)*. Springer Berlin Heidelberg.

Siviy, J., Kirwan, P., Marino, L., & Morley, J. , 2008. *The Value of Harmonizing Multiple Improvement Technologies: A Process Improvement Professional's View.* Software Engineering Institute. Carnegie Mellon University. White paper.

Tregear, R. (2010). *Business process standardization*. In Handbook on Business Process Management 2. Springer Berlin Heidelberg, pp. 307-327.

APPENDIX

Centralization	Interviewee 1	Interviewee 2	Interviewee 3	Interviewee 4	Averages
Index of participation in decision making (5-point Likert-scale)					
How frequently do you usually participate in the decision to hire new staff?	0,50	1,00	0,00	0,25	0,44
How frequently do you usually participate in decisions on the promotion of any of the professional staff?	0,00	1,00	0,00	0,00	0,25
How frequently do you participate in decisions on the adoption of new policies?	0,25	0,75	0,50	0,75	0,56
How frequently do you participate in the decisions on the adoption of new programs?	0,75	0,50	0,00	0,25	0,38
					0,41
Index of hierarchy of authority (4-point Likert-scale)					
There can be little action taken here until a supervisor approves a decision.	0,67	0,67	0,67	0,33	0,58
A person who wants to make his own decisions would be quickly discouraged here.	0,67	0,33	0,33	0,33	0,42
Even small matters have to be referred to someone higher up for a final answer.	0,67	0,33	0,33	0,33	0,42
I have to ask my boss before I do almost anything.	0,33	0,00	0,33	0,33	0,25
Any decision I make has to have my boss's approval.	0,33	0,33	0,67	0,33	0,42
					0,42
Departmental participation in decision making (5-point Likert-scale)					
Employees participate in decisions involving your work.	0,75	0,25	0,75	0,50	0,56
Employees participate in decisions involving their work environment.	0,75	0,75	0,75	0,25	0,63
					0,59
Group average Centralization **0,47**					
Formalization					
Index of Job codification (4-point Likert-scale)					
I feel that I am my own boss in most matters.	1,00	0,33	0,33	0,67	0,58
A person can make his own decisions without checking with anybody else.	0,33	0,67	0,00	0,67	0,42
How things are done here is left up to the person doing the work.	0,33	0,33	1,00	0,33	0,50
People here are allowed to do almost as they please.	0,33	0,33	0,67	0,33	0,42
People here make their own rules on the job.	0,00	0,67	0,67	0,33	0,42
					0,47
Index of rule observation (4-point Likert-scale)					
The employees are constantly being checked on for rule violations.	0,67	0,33	0,33	0,67	0,50
People here feel as though they are constantly being watched to see that they obey all the rules.	1,00	0,67	0,33	0,33	0,58
					0,54
Index of Specificity of job (4-point Likert-scale)					
Whatever situation arises, we have procedures to follow in dealing with it.	0,67	0,33	0,67	0,67	0,58
Everyone has a specific job to do.	0,67	0,67	0,67	0,67	0,67
Going through the proper channels is constantly stressed	1,00	0,67	0,67	0,67	0,75
The organization keeps a written record of everyone's job performance.	0,67	0,67	0,67	0,67	0,67
We are to follow strict operating procedures at all times.	0,67	0,67	0,67	0,67	0,67
Whenever we have a problem, we are supposed to go to the same person for an answer.	0,67	0,67	0,67	0,67	0,67
					0,67
Written communication (5-point Likert-scale)					
The frequency of written communication in your organization is high.	0,75	0,75	0,75	0,50	0,69
					0,69
Group average Formalization **0,59**					

Some Detailed Assessment Results

Physical Data Warehouse Design on NoSQL Databases
OLAP Query Processing over HBase

Lucas C. Scabora[1], Jaqueline J. Brito[1], Ricardo Rodrigues Ciferri[2]
and Cristina Dutra de Aguiar Ciferri[1]

[1]*Department of Computer Science, University of Sao Paulo at Sao Carlos, 13.560-970, Sao Carlos, SP, Brazil*
[2]*Department of Computer Science, Federal University of Sao Carlos, 13.565-905, Sao Carlos, SP, Brazil*
lucascsb@usp.br, jjbrito@icmc.usp.br, ricardo@dc.ufscar.br, cdac@icmc.usp.br

Keywords: Data Warehousing, Physical Design, NoSQL, OLAP Query Processing, HBase, Star Schema Benchmark.

Abstract: Nowadays, data warehousing and online analytical processing (OLAP) are core technologies in business intelligence and therefore have drawn much interest by researchers in the last decade. However, these technologies have been mainly developed for relational database systems in centralized environments. In other words, these technologies have not been designed to be applied in scalable systems such as NoSQL databases. Adapting a data warehousing environment to NoSQL databases introduces several advantages, such as scalability and flexibility. This paper investigates three physical data warehouse designs to adapt the Star Schema Benchmark for its use in NoSQL databases. In particular, our main investigation refers to the OLAP query processing over column-oriented databases using the MapReduce framework. We analyze the impact of distributing attributes among column-families in HBase on the OLAP query performance. Our experiments showed how processing time of OLAP queries was impacted by a physical data warehouse design regarding the number of dimensions accessed and the data volume. We conclude that using distinct distributions of attributes among column-families can improve OLAP query performance in HBase and consequently make the benchmark more suitable for OLAP over NoSQL databases.

1 INTRODUCTION

The comparison among different systems that manipulate huge volumes of data is crucial for modern information systems. Performing analysis over massive volumes of data is a challenge for traditional data warehousing approaches (Chevalier et al., 2015). Data warehouses (DWs) are used for data analysis, in which the data is modeled in a multidimensional schema according to the cube metaphor and on-line analytical processing (OLAP) queries are performed to help the decision-making process. New solutions for big data management are usually implemented on distributed environments, which enables horizontal scalability. Many enterprises use NoSQL (*Not only SQL*) database systems to manage data split in decentralized environments. With the advent of NoSQL systems to store and process data, there is a need to apply systematic techniques for performance comparison, usually conducted by benchmarks.

Benchmarks of DW are tools aimed at answering the question "*Which is the best database system for OLAP query processing?*" (Folkerts et al., 2012). These questions are answered by functional and performance tests, based on properties of each evaluated system. Its goal is to quantify the quality and the performance of a system, in order to make a fair comparison. There are four main requisites of a benchmark (Bog, 2013): relevance, portability, scalability and simplicity. In the context of DW and decision support systems, there are three main benchmarks: TPC-DS (Poess et al., 2002), TPC-H (Moussa, 2012) and Star Schema Benchmark (SSB) (O'Neil et al., 2009). However, they fail in, at least, two requisites. First, they were planned to evaluate relational databases, which can be very different from NoSQL systems, failing on portability. Second, their data generation is centralized and limited, impacting the horizontal scalability of huge data volumes.

There has been a significant amount of work on column-oriented database systems (Abadi et al., 2008). Studies revealed that this type of data storage could support analytical workloads with more than an order of magnitude faster than row-oriented database systems. The performance improvement is related to read-only workload, which reduces the number of I/O operations since most of the queries have to read only the targeted attributes. Based on the as-

Scabora, L., Brito, J., Ciferri, R. and Ciferri, C.
Physical Data Warehouse Design on NoSQL Databases - OLAP Query Processing over HBase.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 111-118
ISBN: 978-989-758-187-8

sumption that, for some specific queries, not all of the values are required at the same time, the column-oriented approach is appropriate for deploying a DW. HBase (George, 2011) is a distributed, persistent and strictly consistent column-oriented NoSQL database system. All data stored in HBase is organized in column-families, as described in Section 2.2. As proposed by Cai et al. (2013), there are an important feature that impact the performance of HBase's read and write operations: the attributes stored in the same or among different column-families.

In this paper, we propose the analysis of OLAP query performance over multiple data organization strategies on HBase. These different strategies refer to different physical DW designs. The present investigation consists in implementing and comparing the performance of OLAP queries over different column-families arrangements. We also highlight scenarios based on different report requirements that could benefit from the designs investigated in the paper. Since a benchmark encompasses the schema and workload of a DW, we tackle the problem by exploring different schemas and analyzing the effects on their workloads.

1.1 Motivating Scenarios

Let a DW storing data related to a shopping corporation represented by the multidimensional data cube. This company is interested in reporting the quantity sold per product per filial per day. OLAP queries issued against this DW depend on the business perspectives of interest. Therefore, we introduce two representative scenarios that motivate the investigations carried out in this paper, as follows:

Scenario 1: The shopping corporation is interested in reporting the daily profit, based on the total of products sold in that day. With this information, the corporation can determine the total income of each month. Other possibility is that the enterprise is focused on the daily profit to perform infrastructure investments. In this scenario, the most frequent OLAP queries access only one dimension of the data cube. We call these queries as one-dimensional queries.

Scenario 2: The corporation is interested in analyzing the amount of units sold of each product over time. More specifically, it is focused on reporting the quantity sold of each product in the last month, or the quantity sold of each product in each filial. In this scenario, the most used queries involve two or more dimensions of the data cube. We call these queries as two-dimensional and three-dimensional queries.

1.2 Contributions

The relevance of our paper is to point out appropriate data organization designs to enhance the performance of OLAP queries in a column-based NoSQL database for different DW enterprise scenarios. We investigate the influence of physical design of the DW schema so that databases administrators can optimize the performance of OLAP queries on distributed column-oriented NoSQL database systems. This paper introduces the contributions described as follows:

1. It proposes a new physical DW design, called FactDate, aimed to improve the performance of one-dimensional queries.
2. It analyses three physical DW designs, each one providing better performance results according to a given scenario. This analysis includes a scalability performance evaluation.
3. It extends the SSB workload by proposing two new OLAP queries, which are used to investigate two-dimensional queries.

The remaining part of this paper is organized as follows. Section 2 summarizes the background, Section 3 reviews related work, Section 4 describes the physical DW designs, including the proposed FactDate design, Section 5 details the queries proposed for SSB, Section 6 addresses the experimental tests, and Section 7 concludes the paper.

2 BACKGROUND

2.1 Data Warehouse and OLAP

Business Intelligence (BI) can be defined as a set of technologies and mechanisms to efficiently extract useful business information from large volumes of data. Nowadays, DWs are inserted in many business information technology applications, enabling the effectively utilization and analysis of information for business planning and decision making (Ciferri et al., 2013). A DW stores data used for analytical tasks to support decision making, such as information about sales, customers and profit. DWs are typically related to the day-to-day company's operations, and can contain millions and even billions of business records.

Following the cube metaphor described in Section 1.1, the DW provides a multidimensional organization of data. When a DW is implemented in relational databases, this organization is usually structured as a star schema, where the fact table stores the measures of the business events and the dimension tables, related to the fact tables, contain the con-

text of these measures (Kimball and Ross, 2013). Star schemas provide better query performance by reducing the number of join operations.

While a DW is considered one of the most used infrastructure for BI systems, OLAP can be interpreted as a front-end analyzing tool. OLAP encompasses complex queries that frequently aggregate and consolidate measures of the fact table, including dimensions perspectives. An example of an OLAP query is: "*How many products were sold by brand and by store in the last year?*", where the measure is the quantity of items sold and the dimensions are brand, stores and date, respectively. In this context, the word analytical refers to extracting information from the DW that is useful to the decision making process, focusing on the analyses of the day-to-day company's operations.

2.2 Column-oriented NoSQL Databases

Column-oriented databases store their data grouped by columns, where the values of each column are contiguously stored on disk. This orientation differs from the row-oriented databases, which store each row entirely and contiguously on disk (George, 2011). As stated in Section 1, an advantage of column-oriented organization is the need of reading only the required attributes of the query, which is the case for OLAP. In a column-oriented database, inserting a tuple requires to write each attribute value of the record separately, raising the number of I/O operations. However, column-oriented storage can improve the performance of queries that access only a subset of columns from a wide table. This occurs because unrequired attributes are not read, reducing the I/O consumption of read-intensive workloads, such as OLAP queries.

An example of column-oriented NoSQL database is HBase, which provides access to each tuple using unique keys called *rownum*. These keys are stored lexicographically. For each attribute of each tuple, HBase stores a cell structure with the following format: <*rownum, cf, column, timestamp, value*>. To retrieve an attribute value, it is required to inform the *rownum*, *cf* (column-family) and *column* fields. Column-families group columns that are stored continuously on disk, in the same file, whose structure is denominated HFile. If a query processes attributes from different column-families, the needed HFiles are joined to reconstruct the query result.

To perform OLAP queries over distributed data, the HBase tables can be used as input to the MapReduce or the Spark frameworks, which are designed for parallel processing of massive datasets (Doulkeridis and Nørvåg, 2014). Companies can implement their own queries using these frameworks, or can integrate HBase with some SQL layers, such as Hive and Phoenix. Hive (Thusoo et al., 2010) is a SQL layer that models an infrastructure of DW, allowing queries to be expressed with a SQL-like language called HiveQL. Phoenix, on the other hand, offers another SQL interface, boosting HBase performance.

2.3 Benchmarking Technique

A benchmarking technique aims to measure performance of an information system and compare it with others. This evaluation consists of performing a set of well-defined tests in order to empirically measure the system's functionality and performance. Moreover, the benchmark must contain a set of operations based on the workload scenario that is going to be tested. Regarding SQL statements, the OLAP query processing can be classified as a read-only workload, focused on select transactions (Bog, 2013).

Regarding benchmarks for DWs, they should encompass four main steps (Floratou et al., 2014): *(i)* schema and workload; *(ii)* data generation; *(iii)* metrics; and *(iv)* validation. In step *(i)*, two issues must be tackled. First, a schema that models a typical application in the domain area of the DW. For the workload, it refers to operations on this schema, represented by OLAP queries with respect to variations of the selectivity. In step *(ii)*, the benchmark defines rules to generate synthetic or real data for the schema, allowing data volumes variations and respecting the selectivity of the workload. In step *(iii)*, some quantitative and qualitative metrics are defined for the benchmark, to report important aspects of the system. Finally, in step *(iv)*, metrics are collected after the workload is applied on the generated data. These metrics are compared to others reported by others databases systems.

3 RELATED WORK

The Star Schema Benchmark (SSB) (O'Neil et al., 2009) is an extension of the TPC-H (Poess and Floyd, 2000), which is designed to measure the performance of analytical queries over database products in support to typical data warehousing applications. SSB implements a genuine star schema, which previous work (Kimball and Ross, 2013; O'Neil et al., 2009) argued that this kind of schema can better represent real-world scenarios. The central fact table of SSB is *LineOrder*, which contains information about sales transactions of a retailer, and this information is stored as different types of measures, like profit, units sold and revenue. Also, SSB defines four dimension tables: *Date*, *Customer*, *Supplier* and *Part*. The SSB

workload is composed of 13 OLAP queries organized in four classes that provide not only functional coverage but also variations of selectivity and hierarchical level used. The main limitation of SSB is the restriction to relational OLAP environments. Further, it also defines queries applied over one, three or four dimension tables, lacking of two-dimensional queries.

The Columnar NoSQL Star Schema Benchmark (CNSSB) (Dehdouh et al., 2014) extends SSB by proposing a benchmark adapted to measure the performance of columnar NoSQL DWs. It defines a schema composed of only one table with several columns, i.e. it denormalizes the SBB's star schema by joining the fact table (*LineOrder*) with the dimensions tables. The attributes of the schema are grouped in column-families (CFs) related to their original dimension. Although this adaptation has an intuitive semantic, the physical design of the schema can influence query performance, and therefore this schema may not be the only one recommended.

Related to the performance evaluation of columnar NoSQL, the work of Cai et al. (2013) measures the HBase performance by using two physical DW designs: (i) only one column-family with multiple columns; and (ii) multiple column-families, where every column-family has only one column. In case (i), reading one row means reading the data of all columns, even if the user does not need them. As a consequence, queries using only a few attributes should consume more I/O and bandwidth. On the other hand, in case (ii), the user needs to read the data separately from each CF and combine them to rebuild the row. When the user request data from a fixed set of columns, storing these specific columns in the same CF should provide a better processing performance than splitting them into several different CFs. The experiments only performed generic read and write operations, without evaluating any aspect of query processing, selectivity, and DW schema.

Another extension of SSB to column-oriented NoSQL databases is proposed by Dehdouh et al. (2015). They introduce three approaches to adapt SSB to HBase, called NLA-SSB, DLA-SSB, and DLA-CF-SSB. NLA-SSB refers to the normalized approach of SSB, while DLA-SSB and DLA-CF-SSB join the fact and the dimension tables. While DLA-SSB groups all dimensions in the same CF, DLA-CF-SSB stores each dimension in a distinct CF, such as CNSSB (Dehdouh et al., 2014). They observed that both DLA-SSB and DLA-CF-SSB did not impact query performance when accessing attributes from different dimensions. However, the experiments only processed a fixed data volume, not analyzing the behavior of query processing as data grow. They also did not evaluate the performance of all SSB's queries, which vary the selectivity and perform aggregations based on real-case enterprise scenarios.

In this section, we addressed the limitations of SSB, and its adaptations, to analyze different physical DW designs on NoSQL databases. In this paper, we tackle these issues by measuring the impact of the CF organization on OLAP queries, considering different data volumes. Further, we propose two new types of queries, which are very important because they allow the investigation of two-dimensional queries.

4 PROPOSED INVESTIGATION

We propose an investigation of the physical DW design on HBase column-oriented NoSQL database, by considering different strategies to arrange attributes into column-families (CFs). Our investigation is motivated by the fact that, as presented in Section 2.2, each CF on HBase is stored in a separated HFile, such that when a query accesses data from two or more CFs, it must read each HFile and join them by the *rownum* field to rebuild the tuple. However, recommendations of HBase state that joining more than two CFs leads to a low performance. As a consequence, it is important to analyze the attribute distribution over CFs regarding the OLAP query context. Depending on the analysis, specifically the number of dimensions aggregated, distinct distributions can benefit or degenerate query performance. Further, different enterprise scenarios may benefit from different physical DW designs. As described in Section 1.1, our work focuses on two enterprise scenarios. Figure 1 depicts the three-level architecture for NoSQL column-oriented databases adopted in our work, showing the adaptations for the physical level regarding distributed NoSQL systems.

By adopting this denormalization, we argue that, at the physical level, the database administrator can decide among three DW designs, as follows:

- Physically organize all attributes in the same CF, as proposed by Cai et al. (2013) and presented as DLA-SSB (Dehdouh et al., 2015). We call this schema as **SameCF**.
- Store each dimension in different CFs, regarding CNSSB (Dehdouh et al., 2014) and DLA-CF-SSB (Dehdouh et al., 2015). We call this schema as **CNSSB**.
- Group some of the more frequently used dimensions to the fact table, which represents the new strategy proposed in this paper. We joined dimension *Date* and call this schema as **FactDate**.

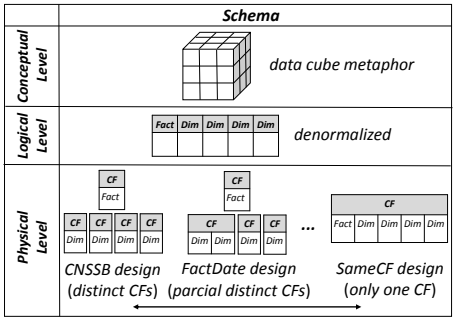


Figure 1: Three-level DW architecture for NoSQL column-oriented databases, with specific adaptations on the physical level that are analyzed in this paper.

First, we implement the CNSSB, which was proposed by Dehdouh et al. (2014) and described in Section 3. Despite the fact that CNSSB is a user understandable organization, any OLAP query that involves at least one dimension of the DW will need to process two CFs, one for the fact's attributes and one for the aggregated dimensions. Our second implementation refers to the SameCF design and consists in storing all the attributes in a single CF, based on the DLA-CF-SSB (Dehdouh et al., 2015) and on the work of Cai et al. (2013). This configuration aims to improve the performance of OLAP queries that analyze a greater number of dimensions, such as Scenario 2 (Section 1.1). Although minimizing the quantity of CFs is a good approach, the organization may decrease the performance of OLAP analysis that uses just a small set of attributes and dimensions.

Finally, we propose a new physical DW implementation, called the FactDate design, which represents an intermediary solution between CNSSB and SameCF, combining the *LineOrder* and *Date* tables. This design is aimed at improving the performance of OLAP queries that use both the fact table and the dimension table *Date*, as most of analytical queries relate their measures to time. This physical design can improve query performance for queries that use both CFs as those described in Scenario 1, i.e. one-dimensional queries. It can also benefit two-dimensional queries, where one of the dimensions used is the dimension *Date*, by decreasing the total number of CFs processed in the query.

5 PROPOSED QUERIES

Another contribution of this paper is the proposal of two new queries to be added to the SSB workload. The need for those queries is to test OLAP aggregations over two dimensions since the SSB workload lacks this type of query. Therefore, the proposed queries are two-dimensional queries. To elaborate queries for a data warehousing benchmark, first we need to determine and vary the selectivity of the queries. The predicates used by OLAP queries defined in SSB have an uniform distribution. Through the cardinality of the attributes, we can define new queries with specific selectivities, similar to the other queries defined by SSB, however performing a two-dimensional analysis. Table 1 defines the cardinality of the predicates used in the proposed queries.

Table 1: Attribute's cardinality of the proposed queries.

attribute	value	attribute	value
d_year	7	*l_quantity*	50
d_yearmonthnum	84	*p_category*	25

The first proposed query, named **Qnew1** (Figure 2), calculates the maximum and minimum revenue, for a given product's category and year, grouped by product's brand. The predicates of this query are defined over the attributes *d_year*, *p_category* and *lo_quantity*, whose combined selectivity is $\frac{1}{7} \times \frac{1}{25} \times \frac{25}{50} = 2.85 \times 10^{-3}$. This value is similar by the same order of magnitude to the queries defined by SSB using other quantities of dimensions, like *Q2.2* (1.60×10^{-3}) and *Q3.2* (1.37×10^{-3}).

```
SELECT p_brand1, max(lo_revenue),
       min(lo_revenue)
FROM lineorder, dates, part
WHERE lo_orderdate = d_datekey
      AND lo_partkey = p_partkey
      AND d_year = 1993
      AND p_category = 'MFGR#11'
      AND lo_quantity < 25
GROUP BY p_brand1 ORDER BY p_brand1;
```

Figure 2: The two-dimensional proposed query **Qnew1**.

The second proposed query is based on a *drill-down* operation over the date hierarchy, changing the analysis from year to month (i.e. attributes *d_year* to *d_yearmonthnum*). This new query, called **Qnew2**, calculates the maximum and minimum revenue for a given product's category and a month of a year, grouped by product's brand. The selectivity of this query is 2.38×10^{-4}, because of the cardinality of the predicate over the attribute *d_yearmonthnum*. Also, this query has the selectivity near to the SSB's queries *Q1.2* (6.49×10^{-4}) and *Q2.3* (2×10^{-4}).

The two new queries are used in the performance evaluation to test the designs detailed in Section 4.

6 PERFORMANCE EVALUATION

In this section, we present the performance evaluation for the three implemented physical designs, CNSSB,

SameCF and FactDate, regarding the execution of the queries of the SSB workload and the queries proposed in Section 5. We also investigate the impact of the data volume scalability. The tests were performed using the following configuration setup:

Hardware: A *cluster* composed of 4 nodes, each node having a quad-core CPU at 3.0 Ghz (i5-3330), 16 GB RAM, 1 TB SATA disk (7200 RPM) and 1 Gb/s network. One node acts only as a dispatcher (*namenode*) and the other three as workers (*datanodes*).

Software: All machines run CentOS (version 7.0). Storage and query processing were performed using HBase (version 0.98.13), Hadoop (version 2.4.1) and ZooKeeper for data partitioning in HBase.

We used the SSB's data generator to generate the dataset, in which the tables were joined into a single denormalized CSV file. This file was first loaded in HBase through conversion to HFile format and then equally distributed among the *datanodes* using MapReduce jobs. Table 2 details, for each Scale Factor (SF), the size of the generated file and the size of the same data after loading it in HBase. Because of the HBase's cell structure, the size of the database is greater than the CSV file size. Each query was implemented using Java (version 1.8) and executed at least five times to collect the average elapsed time.

Table 2: Data volumes used in the experiments.

	Scale Factor (SF)			
	10	**20**	**40**	**80**
CSV (GB)	28	55	109	218
FactDate (GB)	58.4	116.9	233.9	468.2
CNSSB (GB)	58.5	117	234.2	469.9
SameCF (GB)	57.9	115.8	231.8	464.0

6.1 Analyzing the Physical Designs

This experiment evaluates the query processing for the three schemas described in Section 4 using SFs with values of 10 and 20 (Table 2). We organize our discussions considering two aspects: low-dimensional analysis and high-dimensional analysis.

Analysis of Low-dimensional Queries

Regarding Scenario 1, we evaluated OLAP queries involving a few number of dimensions, i.e. we evaluated one and two-dimensional queries. The one-dimensional queries were adapted versions of the SSB query workload. These queries, named *Q1.1*, *Q1.2* and *Q1.3*, depend only on the dimension *Date* and on the fact table. Their differences consist in the predicates involved, which provided different values of selectivity. The two-dimensional queries were the **Qnew1** and **Qnew2** proposed in Section 5.

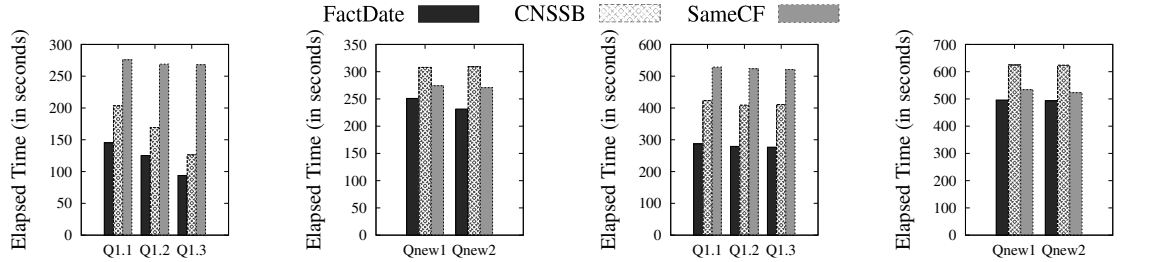
Figure 3 depicts the obtained performance results. Figures 3(a) and 3(c) show that the proposed FactDate outperforms the other designs for one-dimensional queries. FactDate improved the overall performance from 25% to 33% regarding its best competitor, CNSSB. This behavior is justified by the fact that our proposed design processes only one CF, while CNSSB processes two CFs to perform the same query. When comparing the SameCF and the FactDate designs, SameCF contains flatter HFiles because it stores attributes for all dimensions. As a consequence, the one-dimensional query processing requires more time due to larger HFiles. Figures 3(b) and 3(d) illustrate that CNSSB demanded more time to process queries **Qnew1** and **Qnew2**, as it requires accessing three CFs to process them. Also, our proposed FactDate design still outperformed the other designs because it uses only two CFs. Compared to its best competitor, SameCF, FactDate improved query performance from 6% to 14%.

Analysis of High-dimensional Queries

The second part of our analysis evaluated three and four-dimensional queries, which were adapted versions of the SSB workload. These queries are related to Scenario 2, and adapted from SSB's workload. The three-dimensional queries are named *Q2.1* to *Q3.4*, and the four-dimensional queries, *Q4.1*, *Q4.2* and *Q4.3*, accessed all four dimensions of the schema.

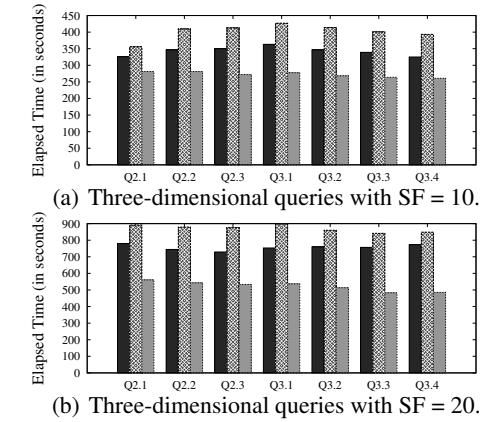
Figure 4 depicts the obtained processing elapsed time. Here, we noticed that processing more than two CFs in the same query provided significant performance losses regarding the FactDate and CNSSB designs. When a query needs to process three or more CFs, the overhead for rebuilding a tuple sharply increased execution time. Figures 4(a) and 4(b) illustrate that SameCF provided the better performance results. They also show that this behavior was maintained when the data volume was increased by two times. Regarding the three-dimensional queries, SameCF improved the overall performance from 14% to 38% when compared to its best competitor, FactDate. Furthermore, when we added more dimensions to the query, the processing time for CNSSB and FactDate increased. They become unsuitable for Scenario 2, as depicted in Figures 4(c) and 4(d). Regarding the four-dimensional queries, the improvement provided by SameCF over FactDate ranged from 47% to 54%.

Both analysis show strong indications about how different enterprise scenarios can require distinct physical DW designs to efficiently attend the most frequent queries. On our next experiment, we analyze how this behavior is related to data volume.



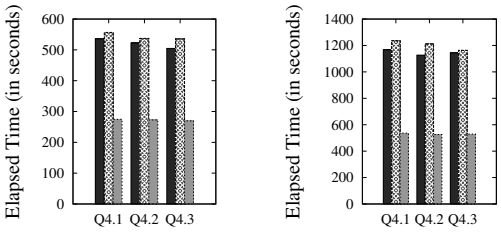
(a) One-dimensional queries with SF = 10. (b) Two-dimensional queries with SF = 10. (c) One-dimensional queries with SF = 20. (d) Two-dimensional queries with SF = 20.

Figure 3: Processing elapsed time for low-dimensional queries, which are related to Scenario 1.



(a) Three-dimensional queries with SF = 10.

(b) Three-dimensional queries with SF = 20.



(c) Four-dimensional queries with SF = 10. (d) Four-dimensional queries with SF = 20.

Figure 4: Processing elapsed time for high-dimensional queries, which are related to Scenario 2.

6.2 Scalability Evaluation

Here, we evaluate query performance, according to the designs described in Section 4, analyzing their behavior as the data volume increases. In this experiment, we used SFs = 10, 20, 40 and 80 (Table 2).

Figure 5 depicts the average processing time for the one-dimensional queries. We can observe that the query performance against SameCF was highly degenerated and produced higher processing times when compared to the other designs. The main difference between FactDate and CNSSB was related to the quantity of CFs, where the queries against FactDate only accessed one CF while the queries against CNSSB accessed two CFs. Regarding FactDate, we observed that reducing the quantity and size of the CFs related to the most frequent queries improved query performance, but if all the dimensions were joined in the same CF, the performance dropped substantially. Further, FactDate boosted the performance by 20% on average when compared to CNSBB.

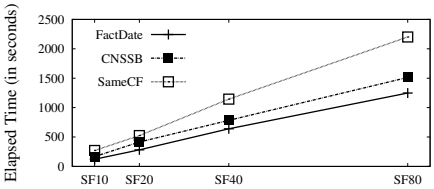


Figure 5: Processing time of one-dimensional queries.

Figure 6 depicts the average processing time for the two-dimensional queries. The three designs showed a similar behavior when processing these queries. However, the proposed FactDate design slightly outperformed the other designs as it processed only two CFs for the two-dimensional queries, while CNSSB design processed three CFs. Comparing FactDate to SameCF, we observed that FactDate deals with two small CFs while SameCF processes only one large CF. Therefore, when two small CFs were joined, FactDate still outperformed the overhead of processing one flatter HFile. This improvement varied from 6% to 11% as the data volume increased.

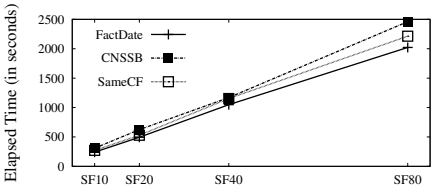
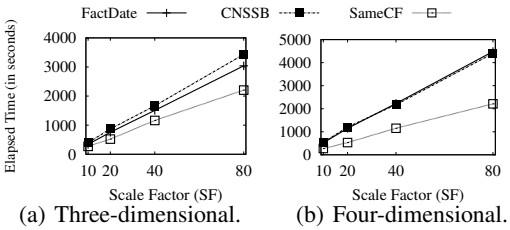


Figure 6: Processing time of two-dimensional queries.

Figure 7(a) shows the average processing time for the three-dimensional queries. For two or more CFs, query performance presents an opposite behavior when compared to the results depicted in Figures 5 and 6. We can observe that grouping all attributes in the same CF boosted the performance of SameCF

over FactDate from 21% to 31% as the data volume increased. Moreover, Figure 7(b) depicts that this improvement, on four-dimensional queries, is up to 54%. We can conclude that, when it comes to processing three or four CFs, FactDate and CNSSB are not suitable for Scenario 2.



(a) Three-dimensional. (b) Four-dimensional.

Figure 7: Processing time of high-dimensional queries.

7 CONCLUSIONS

In this paper, we analyze three physical DW designs, called CNSSB, SameCF, and FactDate. We consider two different enterprise scenarios, determining OLAP queries with different numbers of dimensions. We observe how the attribute arrangement over CFs according to these designs influences OLAP query performance. The results of our experiments showed that storing all data in one CF provided better performance for high-dimensional queries. In this scenario, the SameCF was the most appropriated to be deployed. On the other hand, storing dimensions in different CFs benefited low-dimensional queries. In this scenario, the FactDate and the CNSSB were more appropriated. Further, when processing one-dimensional queries that required data from the dimension *Date*, the FactDate design provided the best performance results. Since data warehousing is characterized by mostly read-only operations, this organization in CFs is an important issue to take into account when comparing NoSQL column-oriented databases.

By using this guideline, the company is able to provide a schema physical design that best suits the most frequent OLAP queries issued against its data warehousing application. Regarding benchmarks, we can conclude that their workload must model different physical designs in order to provide a more accurate evaluation focused on the company interests.

ACKNOWLEDGEMENTS

This work has been supported by the following Brazilian research agencies: FAPESP (Grant: 2014/12233-2), FINEP, CAPES and CNPq.

REFERENCES

Abadi, D. J., Madden, S. R., and Hachem, N. (2008). Column-stores vs. row-stores: How different are they really? In *ACM SIGMOD*, pages 967–980, NY, USA.

Bog, A. (2013). *Benchmarking Transaction and Analytical Processing Systems: The Creation of a Mixed Workload Benchmark and Its Application*. Springer Publishing Company, Incorporated, 1 edition.

Cai, L., Huang, S., Chen, L., and Zheng, Y. (2013). Performance analysis and testing of hbase based on its architecture. In *12th IEEE/ACIS ICIS*, pages 353–358.

Chevalier, M., El Malki, M., Kopliku, A., Teste, O., and Tournier, R. (2015). Implementing Multidimensional Data Warehouses into NoSQL. In *ICEIS*.

Ciferri, C., Ciferri, R., Gómez, L., Schneider, M., Vaisman, A., and Zimányi, E. (2013). Cube algebra: A generic user-centric model and query language for olap cubes. *IJDWM*, 9(2):39–65.

Dehdouh, K., Bentayeb, F., Boussaid, O., and Kabachi, N. (2015). Using the column oriented NoSQL model for implementing big data warehouses. *PDPTA'15*, pages 469–475.

Dehdouh, K., Boussaid, O., and Bentayeb, F. (2014). Columnar NoSQL star schema benchmark. In *MEDI 2014*, pages 281–288.

Doulkeridis, C. and Nørvåg, K. (2014). A survey of large-scale analytical query processing in mapreduce. *The VLDB Journal*, 23(3):355–380.

Floratou, A., Özcan, F., and Schiefer, B. (2014). Benchmarking sql-on-hadoop systems: TPC or not tpc? In *5th WBDB*, pages 63–72.

Folkerts, E., Alexandrov, A., Sachs, K., Iosup, A., Markl, V., and Tosun, C. (2012). Benchmarking in the cloud: What it should, can, and cannot be. In *4th TPCTC*, pages 173–188.

George, L. (2011). *HBase: The Definitive Guide*. O'Reilly Media, 1rd edition.

Kimball, R. and Ross, M. (2013). *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*. Wiley Publishing, 3rd edition.

Moussa, R. (2012). Tpc-h benchmark analytics scenarios and performances on hadoop data clouds. In *NDT*, volume 293, pages 220–234.

O'Neil, P., O'Neil, E., Chen, X., and Revilak, S. (2009). The star schema benchmark and augmented fact table indexing. In *TPCTC*, pages 237–252.

Poess, M. and Floyd, C. (2000). New TPC benchmarks for decision support and web commerce. *SIGMOD Record*, 29(4):64–71.

Poess, M., Smith, B., Kollar, L., and Larson, P. (2002). TPC-DS, taking decision support benchmarking to the next level. In *SIGMOD Conference*, pages 582–587.

Thusoo, A., Sarma, J. S., Jain, N., Shao, Z., Chakka, P., Zhang, N., Anthony, S., Liu, H., and Murthy, R. (2010). Hive - a petabyte scale data warehouse using hadoop. In *26th ICDE*, pages 996–1005.

On the Support of a Similarity-enabled Relational Database Management System in Civilian Crisis Situations

Paulo H. Oliveira, Antonio C. Fraideinberze, Natan A. Laverde, Hugo Gualdron, Andre S. Gonzaga, Lucas D. Ferreira, Willian D. Oliveira, Jose F. Rodrigues-Jr., Robson L. F. Cordeiro, Caetano Traina Jr., Agma J. M. Traina and Elaine P. M. Sousa

Institute of Mathematics and Computer Sciences, University of Sao Paulo, Av. Trabalhador Sancarlense, 400, Sao Carlos, SP, Brazil
{*pholiveira, antoniocf, laverde, asgonzaga, lucas.danfer*}*@usp.br,*
{*gualdron, willian, junio, robson, caetano, agma, parros*}*@icmc.usp.br*

Keywords: Crisis Situation, Crisis Management, Relational Database Management System, Similarity Query.

Abstract: Crowdsourcing solutions can be helpful to extract information from disaster-related data during crisis management. However, certain information can only be obtained through similarity operations. Some of them also depend on additional data stored in a Relational Database Management System (RDBMS). In this context, several works focus on crisis management supported by data. Nevertheless, none of them provide a methodology for employing a similarity-enabled RDBMS in disaster-relief tasks. To fill this gap, we introduce a methodology together with the Data-Centric Crisis Management (DCCM) architecture, which employs our methods over a similarity-enabled RDBMS. We evaluate our proposal through three tasks: classification of incoming data regarding current events, identifying relevant information to guide rescue teams; filtering of incoming data, enhancing the decision support by removing near-duplicate data; and similarity retrieval of historical data, supporting analytical comprehension of the crisis context. To make it possible, similarity-based operations were implemented within one popular, open-source RDBMS. Results using real data from Flickr show that our proposal is feasible for real-time applications. In addition to high performance, accurate results were obtained with a proper combination of techniques for each task. Hence, we expect our work to provide a framework for further developments on crisis management solutions.

1 INTRODUCTION

Crisis situations, such as conflagrations, disasters in crowded events, and workplace accidents in industrial plants, may endanger human life and lead to financial losses. A fast response to this kind of situation is essential to reduce or prevent damage. In this context, software systems aimed at supporting experts in decision-making can be used to better understand and manage crises. A promising line of research is the use of social networks or crowdsourcing (Kudyba, 2014) to gather information from the crisis site.

Several desirable tasks can be performed by software systems designed for aiding in decision-making during crises. One of such tasks is to detect the evidences that best depict the crisis situation, so that rescue teams can be aware of it and prepare themselves properly. For instance, identifying fire or smoke on multimedia data, such as images, videos or textual reports, usually points to conflagration. Some relevant proposals in this direction comprehend fire and smoke detection based on image processing approaches (Celik et al., 2007) and techniques for fire detection designed over image descriptors that focus on detecting fire from social media images (Bedo et al., 2015).

Another important task is to filter the information received from crowdsourcing solutions dedicated to collecting data from crises. When reporting incidents, users might end up sending too much similar information, such as pictures from the same angle of the same object. Such excess of similar data demands a longer time to be processed. Moreover, it turns the decision-making process more time-consuming. Therefore, removing duplicates is an essential task in this context.

The task of searching for similar data in historical databases can support decision-making as well. Take for instance a database that contains images and textual descriptions regarding past crisis situations. If the crowdsourcing system gets, for instance, images depicting fire, a query might be posed on the database to

Oliveira, P., Fraideinberze, A., Laverde, N., Gualdron, H., Gonzaga, A., Ferreira, L., Oliveira, W., Rodrigues-Jr., J., Cordeiro, R., Jr., C., Traina, A. and Sousa, E.
On the Support of a Similarity-enabled Relational Database Management System in Civilian Crisis Situations.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 119-126
ISBN: 978-989-758-187-8

retrieve similar images and the corresponding textual descriptions. Then, based on these results, specialists would potentially infer the kind of material burning in the crisis, by analyzing the color tone of the smoke in the retrieved images and their textual descriptions.

For all those tasks, it is desirable that a commodity system provide functionalities over existing software infrastructure. Commodity systems that can play this role are the Relational Database Management Systems (RDBMS). They are largely available in the current computing technology and are able to bring new functionalities without the need of redesigning the existing software. Moreover, RDBMS provide efficient data storage and retrieval. However, they do not readily support similarity operations, which are needed to address the aforementioned tasks.

Several works in the literature aim at embedding similarity support in RDBMS. Nevertheless, the literature lacks a methodology for employing a similarity-enabled RDBMS in the context of crisis management. This work aims at filling that gap. Our hypothesis is that providing similarity support on an RDBMS helps the decision support in crisis situations.

We contribute with a data-centric architecture for decision-making during crisis situations by means of a similarity-enabled RDBMS. Our proposal is evaluated using an image dataset of real crises from Flickr in performing three tasks:

- **Task 1.** Classification of incoming data regarding current events, detecting the most relevant information to guide rescue teams in the crisis site;
- **Task 2.** Filtering of incoming data, enhancing the decision support of rescue command centers by removing near-duplicate data;
- **Task 3.** Similarity retrieval from past crisis situations, supporting analytical comprehension of the crisis context.

This work has been conducted to cater to demands of the project *RESCUER: Reliable and Smart Crowdsourcing Solution for Emergency and Crisis Management*[1], supported by the European Union's Research and Innovation Funding Program FP7.

The results of our experimentation show that the proposed architecture is effective over crisis scenarios which rely on multimedia data. In addition to the high performance achieved, accurate results are obtained when using a proper combination of techniques.

The rest of the paper is structured as follows. Section 2 presents the related work and Section 3 presents the main concepts for similarity support on RDBMS. Section 4 describes the new Data-Centric Crisis Management architecture, on which the proposed methodology is based. Section 5 presents our methodology, describes the experiments and discusses the results. Finally, the conclusions are presented in Section 6.

[1]http://www.rescuer-project.org/

2 RELATED WORK

Existing research on crisis management highlights the importance of computer-assisted systems to support this task. The approaches may be categorized into different types according to their purpose.

One type refers to localization, whose purpose is to determine where victims are located during a disaster. There are works that accomplish this task by employing cell phone localization techniques, such as International Mobile Subscriber Identity (IMSI) catchers (Reznik et al., 2015).

Another type regards logistics. Examples comprehend an integer programming technique for modeling multiple-resource emergency responses (Zhang et al., 2012) and a methodology for routing rescue teams to multiple communities (Huang et al., 2013).

A different line of work refers to decision-making based on social media (Gibson et al., 2014). Most of the work focus on textual data, specially from services like Twitter (Ghahremanlou et al., 2015).

Although all the aforementioned approaches have been conceived to cater to different requirements, all of them share the characteristic of using Information Communication Technology (ICT) in response to crisis situations. The decision-making systems based on incoming data have one more characteristic: the participation of people somehow involved in the disaster.

Existing work have focused on the importance of crowdsourcing data for crisis management in the post-2015 world (Halder, 2014). Therefore, to describe our methodology, we assume the existence of crowdsourcing as a subsystem dedicated to gathering input data. Additionally, we assume the existence of a command center, where analysts evaluate the input data in order to guide the efforts of a rescue team at the crisis site.

The work of Mehrotra (Mehrotra et al., 2004) is the closest approach with respect to our methodology. That work presents an interesting approach, but it focuses mostly on textual data and spatial-temporal information, rather than on more kinds of complex data, such as images. Furthermore, it lacks a methodology for employing content-based operations. We fill those gaps by providing a methodology to perform such operations over disaster-related data and provide useful information to rescue teams.

3 BACKGROUND

3.1 Content-based Retrieval

Complex data is a common term associated with objects such as images, audio, time series, geographical data and large texts. Such data do not present *order relation* and, therefore, are unable to be compared by relational operators ($<$, $\leq$, $\geq$, $>$). Equality operators ($=$, $\neq$) could be used, but they have little or no meaning when employed on such data. Nevertheless, complex data can be compared according to their content by using *similarity* concepts (Barioni et al., 2011).

The interaction with a content-based retrieval system starts as the user enters a query, providing a complex object as the query example. This complex object is submitted to a feature extractor, which extracts representative characteristics from it and generates a feature vector. The feature vector is sent to an evaluation function, which compares another feature vector stored in the database and returns a value representing the dissimilarity degree (also known as the distance) between both feature vectors. This comparison is repeated over the database, generating the results at the end of the process and sending them to the user.

Two of the most common queries used in content-based retrieval are the Range Query and the k Nearest Neighbor (kNN) Query (Barioni et al., 2011). Range Query is defined by the function $Rq(s_q, \xi)$, where s_q represents an object from data domain $\mathbb{S}$ and ξ is the radius used as distance constraint. The query returns all objects within a distance ξ from s_q. kNN Query is defined by the function $kNNq(s_q, k)$, where s_q represents an object from the data domain $\mathbb{S}$ and k is the number of elements to be returned. The query returns the k most similar objects to s_q. The kNN Queries are employed in the context of Instance-Based Learning (IBL) algorithms, such as the kNN Classifier, which is used in our proposal and thus discussed in Section 3.2.

The feature extraction is usually required because the original representation of a given complex object is not prone to useful and efficient computation. Evaluation functions are able to compute the dissimilarity degree of a pair of feature vectors. These subjects are discussed in Sections 3.3 and 3.4.

The similarity retrieval process can be performed outside an RDBMS. However, enabling an RDBMS with similarity is a promising approach and there are several ways for doing so, as discussed in Section 3.5.

3.2 kNN Classifier

The concept of Instance-Based Learning (IBL) (Aha et al., 1991) comprehends supervised learning algorithms that make predictions based solely on the instances previously stored in the database. In these algorithms, no model is built. The knowledge is represented by the data instances already stored and classified. Then, new instances are classified in relation to the existing stored instances, according to their similarity. One of the main IBL algorithms is the well-known kNN Classifier (Fix and Hodges Jr., 1951).

For a given unlabeled instance, the kNN Classifier retrieves from the database the k most similar instances. Following, it predicts the label based on the retrieved instances, according to some predefined criterion. A simple one is to assign the label of the prevailing class in the k nearest neighbors. Another one is to weigh the retrieved instances by distance, so the closest ones have a higher influence.

3.3 Feature Extractors

One of the main tasks for retrieving complex data by content is the feature extraction process, which maps a high-dimensional input data into a low-dimensional feature space, extracting useful information from raw data while reducing their content. Using proper feature extractors for a complex data domain leads to results closer to what the users expect (Sikora, 2001).

Several feature extractors have been developed for different application domains. The main characteristics investigated in the context of images are color, texture and shape. There are several feature extractors for such characteristics, some of which are part of the MPEG-7 standard (MultiMedia, 2002). In this work, we employ a color-based and a hash-based extractors.

Color-based Extractors. Color-based extractors are commonly used as a basis for other extractors. For this reason, they are the most used visual descriptors in content-based image retrieval. The color-based feature extractors in the MPEG-7 standard are commonly employed in the literature. One of them is the Color Structure Descriptor, which builds a color histogram based on the local features of the image.

Perceptual Hash. An extractor suitable for near-duplicate detection is the Perceptual Hash[2]. It generates a "fingerprint" of a multimedia file derived from various features from its content. These "fingerprints" present the characteristic of being close to one another if the extracted features are similar.

3.4 Evaluation Functions

The dissimilarity between two objects is usually determined by a numerical value obtained from an eval-

[2]http://www.phash.org/

uation function. Objects with smaller values are considered to be more alike.

The Minkowski Family comprehends evaluation functions known as L_p metrics that are widely used in content-based retrieval (Wilson and Martinez, 1997). The L_1 metric corresponds to the Manhattan Distance, also noted as City-Block Distance. The L_2 metric is the well-known Euclidean Distance. Finally, there is the L_∞ metric, also noted as the Chebyshev Distance.

The Hamming Distance (Hamming, 1950), which is another well-known evaluation function, counts the substitutions needed to transform one of the input data into the other. It can be employed in near-duplicate detection tasks, since combining it with the Perceptual Hash leads to accurate results.

3.5 Similarity Support on RDBMS

SimDB (Silva et al., 2010) is a similarity-enabled RDBMS, based on PostgreSQL. The similarity operations and keywords were included in its core. Equivalence rules were also included, which allows alternative query plans. However, similarity queries are only available for numerical data, and traditional queries over other data types are not supported.

SIREN (Barioni et al., 2011) is a middleware between a client application and the RDBMS. The client sends SQL commands extended with similarity keywords, which are checked by SIREN to identify similarity predicates. First, SIREN evaluates the similarity predicates, accessing an index of feature vectors, then uses the RDBMS for traditional predicates.

FMI-SiR (Kaster et al., 2011) is a framework that operates over the RDBMS Oracle, employing user-defined functions to extract features and to index data. MedFMI-SiR is an extension of FMI-SiR for medical images in the Digital Imaging and Communications in Medicine (DICOM) format.

SimbA (Bedo et al., 2014) is a framework that extends the middleware SIREN. SimbA supports the inclusion and combination of feature extractors, evaluation functions and indexes on demand. The queries are processed just like on SIREN.

4 PROPOSED ARCHITECTURE

This section presents our architecture for crisis management, named as Data-Centric Crisis Management (DCCM). Section 4.1 describes the scenario of a typical crisis situation managed by DCCM. Then, Section 4.2 describes our architecture.

4.1 Crisis Management Scenario

Figure 1 shows the scenario of a crisis situation supported by DCCM.

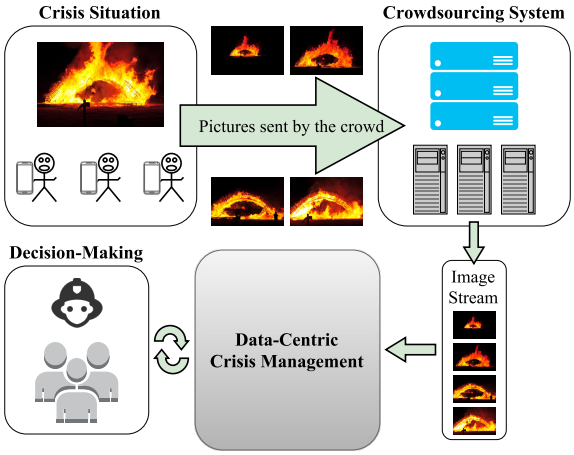


Figure 1: Scenario of a typical crisis situation considering our architecture for crisis management.

In a *Crisis Situation*, eyewitnesses can collect data regarding the event. For instance, they can take pictures, record videos and make textual reports, which are sent to the *Crowdsourcing System*. In Figure 1, the pictures taken by the eyewitnesses are redirected as an *Image Stream* to DCCM. Then, the command center can query DCCM for the *Decision-Making* process.

Additionally, the crowdsourcing system could receive other data, such as metadata (e.g. time and GPS localization) or other data types (e.g. video and text).

4.2 Data-centric Crisis Management

The Data-Centric Crisis Management (DCCM) architecture is represented in Figure 2. The whole mechanism has three processes, each of them depicted in the figure by arrows marked with the letters A, B and C, which represent the tasks introduced in Section 1.

In a crisis situation, we consider the existence of a crowdsourcing system that receives disaster-related complex objects (A1) and submits them to DCCM.

Each object of the data stream is placed in a *Buffer* and analyzed by the *Filtering Engine*. First, the engine checks whether the object is a near duplicate of some other object currently within the *Buffer*. For the near-duplicate checking, the *Filtering Engine* uses the *Similarity Engine* (A2) to extract a feature vector from the object and compare it to the feature vectors of the other objects within the *Buffer*. The object is marked as a near duplicate when its distance from at least another object is at most ξ, which is a threshold defined by specialists according to the application domain.

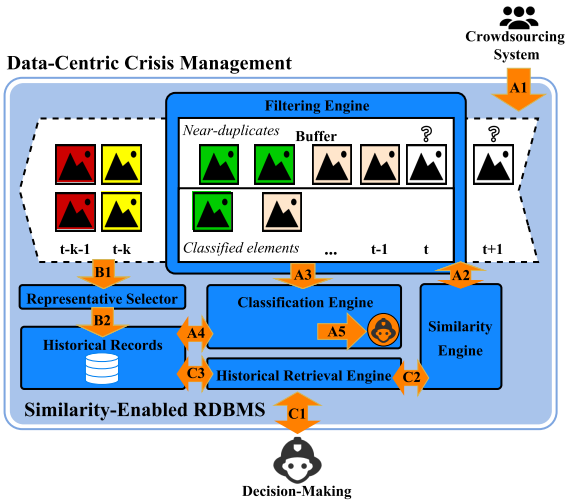


Figure 2: The DCCM architecture, consisting of the tasks: classification (A), filtering (B) and historical retrieval (C).

If the object is not a near duplicate, then it is submitted to the *Classification Engine* (A3). The classification process uses *Historical Records* in a database to train classifier methods (A4). Based on such training, the *Classification Engine* labels the object regarding the event it represents. For instance, it can be labeled as "fire" or "smoke". Finally, the *Classification Engine* notifies the specialists with the now-classified object for the *Decision-Making* process (A5).

If the object is a near duplicate, then it is not submitted to the *Classification Engine*. Instead, it stays in the *Buffer* to be compared to others that come later. Moreover, it is associated with the event of the object of which it is a near duplicate. In Figure 2, the objects from the same event have the same color. The white objects marked with "?" have not been analyzed yet.

The *Buffer* may be determined either by a physical size, such as the number of elements it holds, or by a time window. In Figure 2, it is delimited by a time window of length k, beginning at time t and ending at time $t-k$. The *Buffer* is flushed at every k-th time instant. Before flushing, the *Representative Selector* selects the object of each group that best represents its event (B1) according to a predefined criterion.

If a near-duplicate object is selected as the representative, then it gets the label of the classified object of its group. The already-classified object, in turn, is marked as near duplicate. On the other hand, if the selected representative is already the classified object of its group, then no changes are made. Lastly, the classified objects are stored in the database and the near duplicates are discarded, flushing the *Buffer* (B2).

There is another use case for DCCM, which refers to the historical analyses. The *Decision-Making* team may want to provide complex data samples to retrieve similar events from the past. For this purpose, DCCM provides the *Historical Retrieval Engine* (C1). First, the engine extracts the features from each provided sample (C2). Then, it compares the extracted features against the *Historical Records* and provides its findings to the *Decision-Making* team (C3).

5 CASE STUDY

In this section, we present the case study for evaluating the DCCM architecture over the three tasks discussed earlier. The experiments were carried out over a real crisis dataset known as Flickr-Fire (Bedo et al., 2015) containing 2,000 images extracted from Flickr, 1,000 labeled as "fire" and 1,000 as "not fire".

5.1 Implementation of DCCM

To implement DCCM, we extended the open-source RDBMS PostgreSQL. Our implementation, named as **Kiara**, supports an SQL extension for building similarity queries over complex data (Barioni et al., 2011). Also through the SQL extension, Kiara allows managing feature extractors and evaluation functions, which are dynamically (no recompilation) inserted and updated by user-defined functions written in C++. Kiara makes use of metatables to keep track of feature extractors and evaluation functions associated with the attributes of complex data that a user instantiates.

To support the SQL extension, we built a parser that works like a proxy in the core of Kiara. It receives a query and rewrites only the similarity predicates, expressing them through standard SQL operators. Then, it sends the rewritten queries to the core of Kiara.

After inserting a new extractor, the features are automatically extracted from the complex data (e.g. image, video or text) and then stored in user-defined attributes dedicated to representing such data. Similarity queries can be included through PL/pgSQL functions and new indexes can be included through the interface known as Generalized Search Tree (GiST), already present in PostgreSQL. Moreover, Kiara allows exploring alternative query plans involving traditional and similarity predicates.

5.2 Classification of Incoming Data

5.2.1 Methodology

Classifying disaster-related incoming data is helpful because of two reasons. One of them is to identify the characteristic that best depicts the crisis situation. The other is to store data properly labeled, which improves

further queries on a historical database. To do so, the DCCM architecture employs the *k*NN Classifier.

The parameter k can be selected arbitrarily. However, too small values can be noise-sensitive, whereas too large values allow including more instances from other classes, leading to misclassified instances.

5.2.2 Experimentation and Results

In this task, we classified the elements of the Flickr-Fire dataset. For a robust experimentation, we used the procedure *10-fold cross-validation* for a *k*NN classifier using $k = 10$. We used the Manhattan Distance and the extractor Color Structure Descriptor because existing work showed that they allow accurate results for fire detection (Bedo et al., 2015).

After 10 rounds of evaluation, we took the average accuracy and the average F1 score. The result was the same for both measures, which was 0.86. Considering a real event, this capability would be able to automatically group data according to their content, indicating the main characteristics of the crisis and thus saving the command center crucial time for fast response.

5.3 Filtering of Incoming Data

5.3.1 Methodology

In the task of filtering, we are interested in preventing duplicate information from being classified and subsequently sent to the command center.

To determine whether the incoming data is a near duplicate of existing data, they must be compared by their content. For this purpose, we must employ similarity queries. In this case, though, we are restricted to Range Queries. If the new object is a near duplicate of an object in the buffer, then the distance from each other is at most ξ, which is supposed to be a small threshold (range), since we want to detect pairs of objects that, in essence, represent the same information. Range Queries allow restricting results based on their similarity, differently from *k*NN Queries, which do it by the number of objects retrieved.

Hence, the DCCM architecture prevents near duplicates by using Range Queries. Each object that arrives in the buffer is submitted to a default feature extractor. Then, a Range Query is performed by using the extracted features as the s_q object. The range value ξ must be predefined as well, according to the application domain. If at least one object from the buffer is retrieved by the Range Query, then the s_q object is marked as a near duplicate.

5.3.2 Experimentation and Results

For this experiment, we employed the Hamming Distance with the Perceptual Hash extractor and assumed a buffer size of 80. We filled the buffer with 80 images from Flickr, of which 37 depict "small fire" events and 43 depict "big fire" events. Each of the 80 images was used as the s_q to a Range Query with $\xi = 10$.

The ξ parameter was set to retrieve around half of the images (40 images approximately), in order to be able to return an entire class ("big fire" or "small fire") of images. This allows evaluating the precision of the queries with the Precision-Recall method.

Figure 3 shows the Precision-Recall curve for this set of queries. The curve falls off only after 80% of retrieval. This late fall-off is characteristic of highly effective retrieval methods. In this result, one can notice a precision above 90% up to 50% of recall.

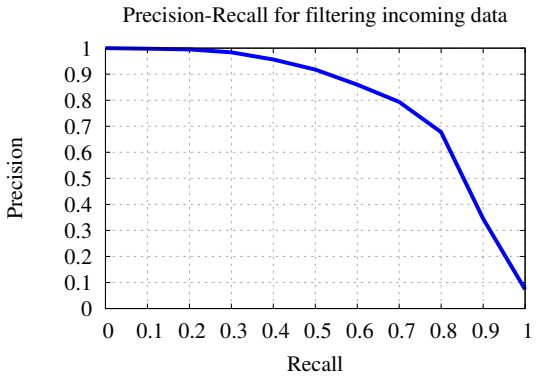


Figure 3: Precision-Recall in the process of filtering incoming data in the buffer.

The results show that DCCM is expected to filter out 90% of near-duplicate data. This is a strong indication that such capability can significantly improve both efficiency and efficacy of a command center. Filtering is the most desirable functionality considered in this work. This is because crowdsourcing is highly prone to produce redundant data. Right after a crisis is installed, if filtering is not possible, the flow of information streamed by eyewitnesses may be too high for the command center to make good use of them. However, a similarity-enabled RDBMS in DCCM is able to handle such situation with basic similarity queries.

5.4 Retrieval of Historical Data

5.4.1 Methodology

In the context of crisis management, the experts from the command center might be willing to analyze data from past events that are similar to the current ones.

Such data may lead to decisions about how to proceed with the current crisis. In these situations, similarity queries play an important role.

Considering the DCCM architecture, this task can be performed whenever the *Decision-Making* experts want information similar to the current data. For every notified data at point A5 of Figure 2, they might provide it as the s_q element to the *Historical Retrieval Engine* in order to get similar information.

5.4.2 Experimentation and Results

For this experiment, we combined the Color Structure Descriptor and the Manhattan Distance.

We performed Range and *k*NN Queries using each of the 2,000 elements from the Flickr-Fire dataset as the s_q element. We set the k parameter to 1,000 and the ξ parameter to 7.2, retrieving an entire class.

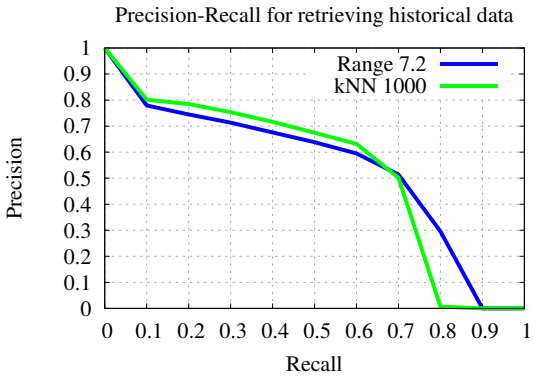


Figure 4: Precision-Recall for retrieving historical data.

We generated the Precision-Recall curve depicted in Figure 4. From these results, one can observe the high precision around 0.8 when fetching 10% of relevant data, roughly 100 images, and around 0.9 when fetching 5%, nearly 50 images — a more realistic scenario. These results point to an effective retrieval of images based on their class.

From the point of view of a command center user, there would be an ample knowledge bank from which initial considerations could be drawn from the current crisis. This initial knowledge has the potential of saving time of rescue actions by preventing past mistakes and fostering successful decisions.

5.5 Overall Performance

Concerning a computer system, it is important to receive the correct response in a timely manner. Therefore, we analyzed the overall performance of DCCM. For this purpose, we carried out one experiment regarding scalability and three regarding the tasks.

Overall Scalability. A solution based on DCCM spends most of its time receiving, storing and indexing data for the sake of similarity retrieval. Therefore, such processing must be efficient. We carried out an experiment to evaluate the time spent extracting features and inserting them into the database. The average time of five rounds is presented in Figure 5.

From the results presented in the figure, one can calculate that the solution is able to process up to 3 images per second — sufficient for most scenarios. These numbers refer to a machine with a hard disk of 5,400RPM; such results could be improved by using SSD disks or RAID subsystems.

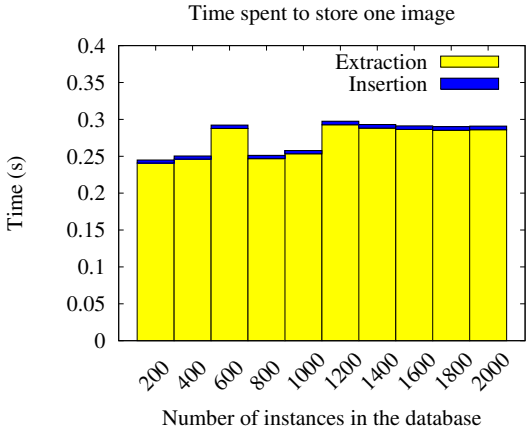


Figure 5: Time to extract features from one image and insert them into the database.

From Figure 5, one can also observe that the time spent inserting images is mostly taken by the feature extraction, while the time for inserting the features remains constant. The extraction time varies according to the image resolutions, which range from 300x214 to 3,240x4,290 pixels in the Flickr-Fire dataset.

Table 1: Overall performance of DCCM over Flickr-Fire.

Task	Average time (sec)
Classification	0.851
Filtering	0.057
Retrieval — Range Query	1.147
Retrieval — *k*NN Query	0.849

Table 1 presents the performance of DCCM to perform the three tasks of our methodology. We ran the classification and filtering tasks 10 times, whereas the retrieval tasks were performed 2,000 times, once for each element in the dataset. For the classification task, we used the distanced-weighted *k*NN classifier, with $k = 10$ and performing 10-fold cross-validation. For the filtering task, we used the 80 aforementioned near-duplicate images and the range value ξ was set to 10. Finally, in the retrieval tasks, ξ was set to 2.8 for the

Range Queries, retrieving around 50 tuples, and k was set to 50 for the kNN Queries. The results, which represent the average time to perform each task once, indicate that our proposal is feasible in a real-time crisis management application.

6 CONCLUSIONS

Fast and precise responses are essential characteristics of computational solutions. In this paper, we proposed the architecture of a solution that can achieve these characteristics in crisis management tasks. In the course of our work, we described the use of a similarity-enabled RDBMS in tasks that could assist a command center in guiding rescue missions. To make it possible, we implemented similarity-based operations within one popular, open-source RDBMS.

The core of our work is related to an innovation project led by the European Union; accordingly, we applied similarity retrieval concepts in an innovative manner, putting together relational and retrieval technologies. To demonstrate our claims, we carried out experiments to evaluate both the efficacy and the efficiency of our proposal. More specifically, we introduced the following functionalities:

- **Classification of Incoming Data.** We proposed to employ kNN classification to classify incoming data, aiming at identifying and characterizing crisis situations faster;
- **Filtering of Incoming Data.** We proposed to employ Range Queries to filter out redundant information, aiming at reducing the data load over the system and over a command center;
- **Retrieval of Historical Data.** We proposed to employ Range and kNN Queries to retrieve data from past crises that are similar to the current one.

The results we obtained for each of these tasks allowed us to claim that a similarity-enabled RDBMS is able to assist in the decision support of command centers when a crisis situation strikes. We conclude by stating that our work demonstrated the use of cutting-edge methods and technologies in a critical scenario, paving the way for similar systems to flourish based on the experiences that we reported.

ACKNOWLEDGEMENTS

This research has been supported, in part, by FAPESP, CAPES, CNPq and the RESCUER project, funded by the European Commission (Grant: 614154).

REFERENCES

Aha, D., Kibler, D., and Albert, M. (1991). Instance-based learning algorithms. *Machine Learning*.

Barioni, M., Kaster, D., Razente, H., Traina, A., and Traina Jr., C. (2011). Querying Multimedia Data by Similarity in Relational DBMS. In *Advanced Database Query Systems*.

Bedo, M., Blanco, G., Oliveira, W., Cazzolato, M., Costa, A., Rodrigues Jr., J., Traina, A., and Traina Jr., C. (2015). Techniques for effective and efficient fire detection from social media images. ICEIS '15.

Bedo, M., Traina, A., and Traina Jr., C. (2014). Seamless integration of distance functions and feature vectors for similarity-queries processing. *JIDM*.

Celik, T., Ozkaramanli, H., and Demirel, H. (2007). Fire and smoke detection without sensors: Image processing based approach. EUSIPCO '07.

Fix, E. and Hodges Jr., J. (1951). Discriminatory analysis — Nonparametric discrimination: Consistency properties. Technical report, DTIC Document.

Ghahremanlou, L., Sherchan, W., and Thom, J. (2015). Geotagging Twitter messages in crisis management. *The Computer Journal*.

Gibson, H., Andrews, S., Domdouzis, K., Hirsch, L., and Akhgar, B. (2014). Combining big social media data and FCA for crisis response. UCC '14.

Halder, B. (2014). Crowdsourcing collection of data for crisis governance in the post-2015 world: Potential offers and crucial challenges. ICEGOV '14.

Hamming, R. W. (1950). Error detecting and error correcting codes. *Bell System Technical Journal*.

Huang, M., Smilowitz, K., and Balcik, B. (2013). A continuous approximation approach for assessment routing in disaster relief. *Transportation Research Part B*.

Kaster, D., Bugatti, P., Ponciano-Silva, M., Traina, A., Marques, P., Santos, A., and Traina Jr., C. (2011). MedFMI-SiR: A powerful DBMS solution for large-scale medical image retrieval. ITBAM '11.

Kudyba, S. (2014). *Big Data, Mining, and Analytics: Components of Strategic Decision Making*.

Mehrotra, S., Butts, C., Kalashnikov, D., Venkatasubramanian, N., Rao, R., Chockalingam, G., Eguchi, R., Adams, B., and Huyck, C. (2004). Project Rescue: Challenges in responding to the unexpected. EI '04.

MultiMedia, I. (2002). MPEG-7: The generic multimedia content description standard, p. 1. *IEEE MultiMedia*.

Reznik, T., Horakova, B., and Szturc, R. (2015). Advanced methods of cell phone localization for crisis and emergency management applications. *IJDE*.

Sikora, T. (2001). The MPEG-7 visual standard for content description — An overview. *IEEE Trans. Cir. Sys. Vid.*

Silva, Y., Aly, A., Aref, W., and Larson, P. (2010). SimDB: A similarity-aware database system. SIGMOD '10.

Wilson, D. and Martinez, T. (1997). Improved heterogeneous distance functions. *J. Artif. Int. Res.*

Zhang, J., Li, J., and Liu, Z. (2012). Multiple-resource and multiple-depot emergency response problem considering secondary disasters. *Expert Syst. Appl.*

4D-SETL
A Semantic Data Integration Framework

Sergio de Cesare[1], George Foy[2] and Mark Lycett[2]
[1]*College of Business, Arts and Social Science, Brunel University London, Uxbridge, U.K.*
[2]*College of Engineering, Design and Physical Sciences, Brunel University London, Uxbridge, U.K.*
{sergio.decesare, george.foy, mark.lycett}@brunel.ac.uk

Keywords: Foundational Ontology, Perdurantist, 4D, Semantic Data Integration, Modelling, Graph Databases, Integration Frameworks.

Abstract: Although successfully employed as the foundation for a number of large-scale government and energy industry projects, foundational ontologies have not been widely adopted within mainstream Enterprise Systems (ES) data integration practice. However, as the closed-worlds of ES are opened to Internet scale data sources, there is an emerging need to better understand the semantics of such data and how they can be integrated. Foundational ontologies can help establish this understanding and therefore, there is a need to investigate how such ontologies can be applied to underpin practical ES integration solutions. This paper describes research undertaken to assess the effectiveness of such an approach through the development and application of the 4D-Semantic Extract Transform Load (4D-SETL) framework. 4D-SETL was employed to integrate a number of large scale datasets and to persist the resultant ontology within a prototype warehouse based on a graph database. The advantages of the approach included the ability to combine foundational, domain and instance level ontological objects within a single coherent system. Furthermore, the approach provided a clear means of establishing and maintaining the identity of domain objects as their constituent spatiotemporal parts unfolded over time, enabling process and static data to be combined within a single model.

1 INTRODUCTION

An enterprise may acquire data from many sources in many different forms (Zikopoulos and Eaton, 2011). Key considerations in integrating such data include dealing with the diversity of representation and the interpretation of the inherent explicit and implicit semantics. The latter of these considerations is particularly important in the context of ES integration as, if left unrecognised, it can lead to the things of importance (e.g., domain objects and their relationships), their nuances and the state of affairs they represent being misinterpreted (Lycett, 2013). These considerations are well recognised within database integration projects (Arsanjani, 2002; Campbell and Shapiro, 1995; Sheth and Larson, 1990).

Ontology has emerged as a promising way of dealing with such diversity, however many popular domain ontologies have no grounding in a consistent foundational view of reality (Cregan, 2007) and therefore can add further diversity. A foundational ontology can be employed to provide this 'grounded' view of reality and thus provide an explicit theory and a common reference through which to interpret, model and thus integrate data. Foundational ontology "defines a range of top-level domain-independent ontological categories, which form a general foundation for more elaborated domain-specific ontologies" (Guizzardi *et al.* 2008). From a philosophical perspective, foundational ontologies provide the criteria for ontological commitments – statements on the things believed to exist within the context of a particular theory (Bricker, 2014). Several foundational ontologies currently exist (Gangemi *et al.,* 2002; Grenon and Smith 2004; Partridge 2005; Guizzardi, *et al.,* 2008; Herre 2010) which differ in the ontological commitments they make but, importantly, there is little existing work that examines their suitability as an ultimate 'mediating layer' within a practical data integration context.

Here, we employ a 4D foundational ontology as a means of dealing with the diversity of representation and semantics within acquired data. We do this in the context of a semantic Extract-Transform-Load framework (called 4D-SETL from

Cesare, S., Foy, G. and Lycett, M.
4D-SETL - A Semantic Data Integration Framework.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 127-134
ISBN: 978-989-758-187-8

this point) that uses a 4D foundational ontology to harmonise data, then generates a graph database that accords with the semantic commitments made by the ontology. We examine the effectiveness of the framework by applying it to semantically interpret and integrate a number of large-scale datasets and to instantiate a data warehouse based on a graph database to persist the resultant ontology. In doing this, the paper is structured as follows. Section 1 outlines the problem of variety in terms of the semantic heterogeneity that exists within systems modelling and foundational ontologies and also identifies a number of the weaknesses in current integration approaches. Section 2 describes the core categories and foundational patterns of the BORO foundational ontology. Section 3 introduces the 4D-SETL framework. Section 4 provides details of the experimental dataset integrated. Section 5 details the outcomes and limitations.

1.1 Semantic Data Integration

Data integration is problematic on several counts. Firstly, people perceive and conceptualise reality in different ways. Even when a set of models is developed by the same individual, they can make different (and sometimes arbitrary) choices about the same reality at different times and in different contexts (Kent, 1978). Secondly, in the course of modelling reality, a designer may confuse what is being represented with the representation itself (Partridge *et al*., 2013). Thirdly, different structures and restrictions are introduced by heterogeneous modelling methods and languages (e.g., Entity-Relationship, OWL etc.). Fourthly, it is common practice to develop a number of models in systems development – conceptual, logical and physical data models for example (Codd, 1970). This layering can have an adverse effect as the original semantic structures may be distorted or lost completely as the emphasis of the modelling activity moves from representing the real world to representing data structures. Consequently, when integrating data that originates from different sources, the problem of semantic heterogeneity arises – resolution is required regarding differences in meaning, interpretation or the intended use of related data which forms a barrier to coherent semantic data integration (Doan, Noy and Halevy, 2004).

1.2 Heterogeneous Foundational Ontologies

Ontology provides a way of dealing with semantic data integration. From a computational standpoint, an ontology is generally taken as a 'specification of a conceptualization' (Gruber, 1995) – that is, a description of the concepts and relationships that are considered legitimate within a particular system of thought. In terms of the concrete implementation of software systems, foundational ontologies can be used to establish the fundamental 'meta' objects and relations used to construct more specific domain ontologies. If a common foundational theory is extended and specialised to model a number of domain ontologies, then objects common to each of these domains will have the necessary (common) grounding to enable semantic integration. Consequently, foundational ontologies are important as they provide a standpoint that underpins all the domain models to be integrated – providing a semantic grounding.

It is the case, however, that several such standpoints (related to foundational ontologies) exist, Each provides a criterion for the ontological commitments made (implicitly or explicitly), which are principally the things believed to exist within the context of a particular theory such as four-dimensionalism (Quine, 1952; Sider, 2003). An understanding of ontological commitment, however, means that the computational view of ontology needs to defer to a philosophical one, which is more specifically concerned with the nature of being (metaphysics). As metaphysical theories differ on a number of dimensions (realism versus idealism, endurantism versus perdurantism to name but two) differences thus appear in foundational ontologies. Furthermore, and perhaps more importantly, the degree to which foundational ontologies are actually grounded in metaphysics varies. Clearly, a lack of consensus at the metaphysical level introduces obstacles to semantic integration (Campbell and Shapiro, 1995) that result in weaknesses in computational applications:

a) **Lack of Grounding**. Many current models employed within information systems have no form of grounding in a more fundamental theory (Cregan, 2007). Thus the ontological commitments underlying the model are unknown. On examination of many Linked Open Data ontologies, they are often ungrounded.
b) **Integrating Elements from Models which are Founded on Different Theories.** There are many automatic translation techniques for translating RDBMS schema and data to an OWL 'ontology'. However, there is a lack of recognition that the expressivity of Description Logics (that underlie OWL) and RDBMS are

different as are the unique naming and the open/closed world-view assumptions.

c) **Model Strata and Translations.** As noted earlier, the requirement to translate the high-level models of reality created at the initial design to structures that are focused on the execution environment can result in semantic distortion. There is also the problem of translating between run-time representations; the often cited OO-RDBMS impedance mismatch (Ireland *et al.*, 2009).

d) **Over Simplification to Fit a Model of Reality to a Tractable First Order Logic (FOL) Theory.** The simplification of the abstraction of reality to fit neatly into a FOL theory, thus ignoring the fact that reality is not so simple and higher order objects exist (Bailey, 2011).

e) **Dividing Models into Static and Dynamic Types.** The separation of static and dynamic aspects of reality into different structural and process models leads to the development of incompatible abstractions together with 'exotic' relations that are employed to bridge these static and dynamic worlds.

f) **Naming and Meaning Confusion.** There is often confusion between an entity's naming and meaning (Bailey and Partridge, 2009). An object's place in reality (and within ontology) should define its meaning.

g) **Establishing Identify.** Many modelling and information systems use ephemeral means of establishing an entity's identity which do not function well over time.

h) **Employing Techniques that do Not Scale.** Software tools such as OWL tableau calculus-based reasoners are constrained by memory and cannot be easily scaled to inference over ontologies containing large instance populations (Bock *et al*., 2008). The alternative is to use simplified semantics and rule based reasoning - that could in many cases employ standard RDBMS techniques.

i) **Semantic Integration Mismatches**. For a more extensive discussion on the types of semantic integration mismatches see Visser *et al.*, (1997) who provide an extensive list of e semantic mismatches that can occur when integrating disparate datasets.

2 BORO FOUNDATIONAL ONTOLOGY

Having examined several foundational ontologies from a philosophical perspective, the research described here adopts the Business Object Reference Ontology (BORO) (Partridge, 2005) to semantically interpret the original datasets and models. We adopt BORO on the grounds that the ontology can overcome the dichotomy that exists between dynamic and static modelling paradigms and its metaphysical thoroughness. Hence, the same model can represent processes and things that are not traditionally considered as processes (e.g., people, products, machines, etc.). BORO represents all individual elements (e.g. the activity, the person assuming a role and the resource consumed) in exactly the same way (i.e. as spatiotemporal extents). BORO is based on a philosophical (rather than computational) definition of ontology because it requires more clarity on "the set of things whose existence is acknowledged by a particular theory or system of thought" (Lowe, 1998, p.634.). Key to overcoming the dichotomy noted is the fact that BORO is perdurantist (and thus extensionalist) in its nature. In perdurantism (or 4D) an individual object is never wholly present at one point is time, but only partly present (a temporal part). For example, John is not fully present in any given phase of his life (e.g., childhood), he is fully present from his birth to his death only – therefore, John's childhood is a temporal part of John. Identity is thus defined by an individual object's spatiotemporal extension (or extent). Figure 1 represents the key part of the foundational ontology relevant for the purposes of this paper.

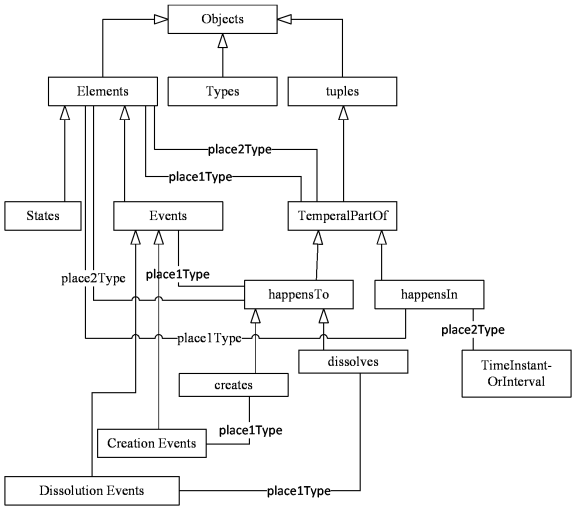


Figure 1: BORO Foundational Ontology (top level).

More in-depth discussions are provided in Partridge, 2002; 2005; Bailey and Partridge 2009; Bailey, 2011; Partridge et al., 2013). At the top level

the BORO foundational ontology represents:

• **Elements**, which are individual objects or objects with a spatiotemporal extent. For example, the person *John*.

• **Types**, which are sets or objects that can have instances. The identity of a type is also extensional but, in this case, it is defined as the set of its instances (i.e. members). For example, the extension of the type *Persons* is the set of all people.

• **Tuples**, which are relationships between objects. The identity of a tuple is defined by the places in the tuple. An example is (*Persons*, *John*) in which the type Persons and the element John occupy places 1 and 2 in the tuple respectively. This specific tuple is an instance of the tuple type *typeInstances* in BORO.

In turn, Elements is subtyped by:

• **Events:** An event is an element that does not persist through time (i.e. an event has zero 'thickness' along the time dimension). Events represent temporal boundaries that either create (CreationEvents) or dissolve (DissolutionEvents) elements (e.g., a person or a person's childhood state).

• **States:** A state is an element that persists through time. States (and elements in general) are bounded by events. A state (like all elements) can have further temporal parts (i.e. states and events). Specific TupleTypes (or types whose instance are tuples) relevant here are:

• **temporalPartOf:** This tuple type relates an individual with its temporal parts (states and/or events).

• **happensTo:** This tuple type relates an event with one or more elements affected by the event. happensTo has two subtypes:
- creates: Relates a creation event with the element(s) whose creation is triggered by the event.
- dissolves: Relates a dissolution event with the element(s) whose dissolution is triggered by the event.

• **happensIn:** This tuple type relates an event with a time instant or interval (TimeInstantsOrIntervals) and it indicates the time in which an event takes place.

As a note of importance for the example shown later, names are types in BORO. The instances of the name of an individual (e.g. John Smith's Name) are all utterances (written, spoken, etc.) that name that individual (e.g., John Smith). Therefore while a name, is a type its instances are spatiotemporal extents. To provide clarity within the ontology, 'names' as much elements of the ontology as the things they name. A name object will belong to a Name Space which holds all names related to a particular naming authority or domain. As the ontology adopts a theory of utterances – each utterance of a name is an individual element and so has an extent (Strawson, 1964). Therefore, a name is a Type that has as instances all utterances of the same name individuals.

3 4D SEMANTIC EXTRACT TRANSFORM AND LOAD FRAMEWORK (4D-SETL)

Given an outline understanding of the foundational ontology, we now describe a Semantic Extract-Transform-Load framework. Given a variety of data input, 4D-SETL is designed to output a graph database in accordance with the BORO foundational ontology. The framework was designed around a number of industry standard tools and technologies (e.g., a UML design tool and a Graph Database), supplemented where necessary with custom software implemented in Java. The key technology choices made for the initial implementation were threefold. First spreadsheets were employed to document each dataset. Second, a UML design tool (Enterprise Architect) was selected as the graphical design tool for the ontological models and a BORO custom UML profile created: The advantage is that BORO UML enables easy manipulation and design of each of the required domain ontologies. Last, the Neo4J Graph database was chosen for persistence, for several reasons: (a) Primarily due to its flexibility in enabling BORO to be used as the foundational 'schema' (both can be seen as graphs); (b) scalability in order to handle model and instance data volume appropriately; (c) Neo4J's web-based interface also provides access to the graph database for development testing; and (d) Neo4J Cypher provides an appropriate means of querying and updating the graph database resident data.

The Semantic Extract Transform Load (ETL) process is shown in Figure 2, the key stages of the process are as follows:

a) **Semantic Extraction and Transformation.** The input data to a semantic integration process may be structured in many forms –e.g., as fixed record or delimited tabular files, RDF, RDFS, OWL etc. – and may consist of both model (schema) level and/or instance level data. Thus the first stage in the semantic integration

process begins with documenting the dataset which can be considered a semantic extraction and transformation process. The BORO foundation provides a view of reality and the patterns that can be employed perform this interpretation and transformation. The foundational ontology provides the equivalent of a canonical data model (Saltor *et al*., 1991) that can be employed to develop domain models providing the semantics that are common to all datasets that will be integrated. Thus the translation process results in a new schema (domain ontology) that extends the ontic categories and patterns of the foundation. Through this process, schemas are developed to represent the entities and relationships that are represented by the data. Finally this schema is documented using a profile of UML that conforms to BORO semantics.

b) **Ontology Model ETL.** Once a domain model has been created in an ontologically consistent form the semantic load and integration process can be undertaken. Firstly the domain ontological model, which includes such patterns as type and classification taxonomies, is translated from the BORO UML model and loaded to the graph database. The 4D-SETL framework provides a Java application to translate the BORO UML and load it to the graph database.

c) **Ontology Data ETL.** Next, the instance level dataset is loaded and integrated. It is through this process that the integration of individual elements takes place. Integration can be considered to take place within vertical and horizontal planes. Initially the 'vertical' relationships between an individual element and the domain ontology (and hence the foundation ontology) is asserted, which consists of establishing the individual relationships (such as type instance, etc.). Then the 'horizontal' relationships that are deemed to hold between individual domain level objects are established (such as a company being located at a particular geographic location). Foundational ontological patterns can then be applied to simplify this process. This can be a complex process that requires both one-to-many and many-to-one transformations. The 4D-SETL framework provides a Java application to perform this process.

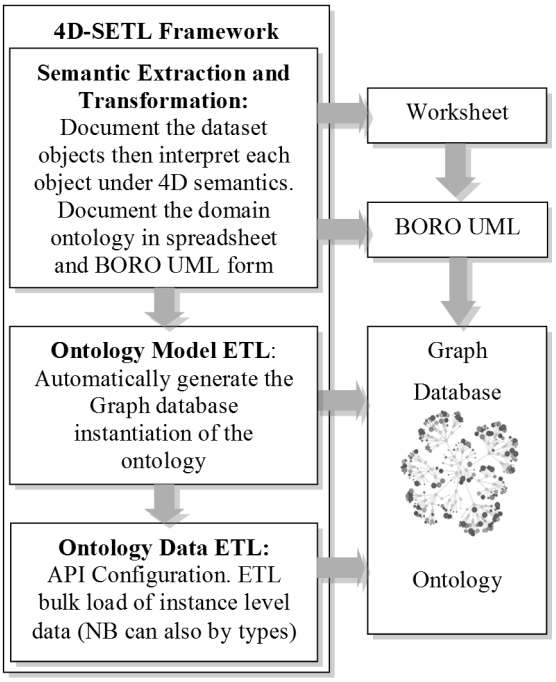


Figure 2: 4D-SETL Framework.

4 EXPERIMENTAL DATA

As the foundational ontology is an integral part of the framework, prior to processing any of the domain ontological elements (model and datasets) the foundational ontology is transformed to graph format and loaded to the database. This is achieved via the 4D-SETL framework which extracts the BORO UML as XML (XMI), then transforms it to a set of nodes and edges that are loaded to the database. The graph database ontology also includes the UML model identifiers as indexed node and edge parameter key-values, these are employed to enable the reproduction of the design time UML models within the graph database runtime environment and to establish the relationships between the foundation and other subsequent domain model elements that are loaded. The 4D-SETL framework was applied to Extract, Transform and Load (ETL) five datasets of varying scale and complexity related to corporate data:

- Calendar: temporal locations (1856 to present).
- Location: spatial locations (~2.5M locations).
- Standard Industrial Classification (taxonomy).
- UK Companies (~3.5M)
- UK corporate officers (~12M)

5 OUTCOMES AND LIMITATIONS

Having applied the framework, our experience is that BORO provides a coherent lens through which to view and model the world together with the foundational ontological elements and patterns through which the domain ontologies can be developed to represent the datasets to be semantically integrated (Partridge, 2002). In terms of domain ontology development, this work concurs with the work of Keet (2011), who stated that employing foundational ontologies provides advantages in terms of the quality and interoperability of domain ontologies. Developing such domain ontologies provided the means of semantically integrating data conforming to different models and theories – a necessary evil in dealing with variety in big data.

Employing a graph database provided the means of importing and restructuring data in a manner that directly reflects the ontological model patterns without the normal translation to tabular RDBMS or Object Oriented form and not introducing the 'impedance mismatch' problem (Ireland *et al.*, 2009). Dispensing with RDBMS storage in favour of a property graph data model removed the partitioning of the storage structures between data and schema and allows both 'schema' ontological model objects and instance level objects to be updated at run time. This supports the work related to graph databases by Webber (2012). Related to this finding, it was also demonstrated in this study that patterns could be established within the warehouse that directly reflected the physical or socially constructed patterns of reality such as taxonomies and taxonomic ranks, the latter of which employed the powertype pattern (equivalent to the set theoretic powerset) to more accurately reflect the nature of such classification systems. These aspects of 4D ontologies (along with others) provide a greater level of flexibility and reusability when evolving the warehouse system and therefore concur and take forward the initial findings of Partridge (2002).

In practical terms, we propose that the data structures resulting from the 4D-SETL process are more suitable for discovering relationships within data rather than for example processing aggregate data (Vicknair *et al*., 2010). It is relatively easy, for example, to discover all relationships that exist between two elements using a standard algorithm from the Neo4J library (designed to find all available paths or the shortest path between two nodes). Further, the Cypher graph database query facilities provide the means of discovering more complex patterns of relationships between the people, company officers, company activities, events and physical location. Finally, it was found through the evaluation and empirical experiment on the prototype warehouse (graph database) that data load and information retrieval response times that the prototype could be developed into a practical information system. This was confirmed by performing test data query (graph traversal) experiments that for example, performed graph traversals to retrieve all companies within a postcode location (61 milliseconds) and all officers for a specific business organisation (37 milliseconds) thus the prototype produced indicative response times within bounds that would support interactive applications (Bhatti *et al.,* 2000). Testing also confirmed the graph database performance evaluation undertaken by Vicknair *et al.* (2010). Thus using a graph database and the parameter graph model to store the ontology, alongside query information via graph traversal, circumvents the issues that limit the ability of systems built using triple stores and tableau calculus-based reasoner technology to deal with ontologies that are both expressive and have with very large instance level elements (arguably exactly what one would want from big data). Neo4J is highly scalable and provides capacities for Nodes/Edges of ~34 billion and properties at least ~ 68 billion respectively.

With the issue of disparate data sources in mind, the work here has: (a) Examined the potential contribution of foundational ontology; and (b) described an implementation of a Semantic Extract-Transform-Load framework (4D-SETL) based on BORO, a 4D foundational ontology. Foundational ontologies provide a 'grounding' for our view of reality and thus provide a common reference through which to model and integrate heterogeneous data. The 4D-SETL framework uses the BORO foundational ontology to harmonise data and then generates a graph database that accords with the semantic commitments made by that ontology. The effectiveness of the framework was examined applying it to large-scale open datasets related to company information to semantically interpret and integrate the datasets and to instantiate a prototype graph database warehouse to persist the resultant ontology. Our implementation is a prototype at this stage and the use of foundational ontologies is not without challenge (e.g., automation in the context of real-time data streams). Accepting such limitations, however, the potential utility of the 4D-SETL framework can be seen in its ability to model and

instantiate a number of complex ontological structures, such as higher order taxonomic ranks. The patterns specialised from the core foundational BORO ontology patterns offer a high degree of flexibility and reusability when evolving the graph-based warehouse system. We have thus demonstrated how a 4D (perdurantist) foundational ontology can be employed to semantically interpret and structure data, showing that a single coherent ontology can be developed and loaded to a graph database without the problems associated with current approaches – e.g., model distortion, over simplification or scalability problems.

Understandably, the work here is not without its limitations, which may be summarised as follows. First, and at the outset, the interpretation process is manual. BORO encourages the development of patterns (for ontological reuse), which allow for partial automisation but skills in ontological modelling are necessary throughout. In the context of dealing with variety in big data automatic translation of data is of particular importance. As a consequence, pattern development and the extraction of the rules associated with that are also of importance for ongoing research. Second, as previously noted, BORO is one of several foundational ontologies and further work is required to understand their relative comparative advantages and disadvantages.

The work here was supported by funding from the Engineering and Physical Sciences Research Council (Project EP/L021250/1). The experimental research data and metadata (Ontology) for this project was sourced from the following organisations: Companies House (2016), Company Information; UK Office of National Statistics (2016) Geographic Location (ONSPD Product); UK Office of National Statistics (2016), Standard Industrial Classification; Company Officers: (A commercial credit reference agency); BORO Engineering Limited (2016), Foundational Ontology.

The Companies House and ONS Datasets are UK Open Government Data and can be freely downloaded. The Company Officers and BORO Ontology are commercial in-confidence.

REFERENCES

Arsanjani, A. (2002) 'Developing and Integrating enterprise Components and Services', *Communications of the ACM,* 45(10), pp. 30-34.

Bailey, I. (2011) 'Enterprise Ontologies–Better Models of Business', in *Intelligence-based systems engineering.* Springer, pp. 327-342.

Bailey, I. and Partridge, C. (2009) 'Working with extensional ontology for defence applications', *Ontology in Intelligence Conference.*

Bhatti, N., Bouch, A. and Kuchinsky, A. (2000) 'Integrating user-perceived quality into web server design', *Computer Networks,* 33(1), pp. 1-16.

Bock, J., Haase, P., Ji, Q. and Volz, R. (2008) 'Benchmarking OWL reasoners', *Proc. of the ARea2008 Workshop, Tenerife, Spain (June 2008).*

BORO Engineering Limited (2016) 'BORO Ontology'. Available from: < http://www.borosolutions.co.uk/solutions/resources/boro-presentations-and papers >. [16 February 2016].

Bricker, P. (2014) 'Ontological Commitment', in Edward N. Zalta (ed.) *The Stanford Encyclopedia of Philosophy.* Winter 2014 edn.

Campbell, A. and Shapiro, S. (1995) 'Ontological Mediation: An Overview', *IJCAI Workshop on Basic Ontological Issues in Knowledge Sharing.* 1995. AAAI Press, Menlo Park, CA,.

Codd, E. (1970) 'A relational model of data for large shared data banks', *Communications of the ACM,* 13(6), pp. 377-387.

Companies House (2016), Free Company Data Product. Available from: < http://download.companieshouse.gov.uk/en_output.html >. [16 February 2016].

Cregan, A. (2007) 'Symbol grounding for the semantic web', in *The Semantic Web: Research and Applications.* Springer, pp. 429-442.

Doan, A., Noy, N.F. and Halevy, A.Y. (2004) 'Introduction to the special issue on semantic integration', *ACM Sigmod Record,* 33(4), pp. 11-13.

Gangemi, A., Guarino, N., Masolo, C., Oltramari, A. and Schneider, L. (2002) 'Sweetening ontologies with DOLCE', in *Knowledge engineering and knowledge management: Ontologies and the semantic Web.* Springer, pp. 166-181.

Grenon, P. and Smith, B. (2004) 'SNAP and SPAN: Towards dynamic spatial ontology', *Spatial cognition and computation,* 4(1), pp. 69-104.

Gruber, T.R. (1995) 'Toward principles for the design of ontologies used for knowledge sharing?', *International journal of human-computer studies,* 43(5), pp. 907-928.

Guizzardi, G., de Almeida Falbo, R. and Guizzardi, R. (2008) 'Grounding Software Domain Ontologies in the Unified Foundational Ontology (UFO): The case of the ODE Software Process Ontology.', *CIbSE.* , 127-140.

Herre, H. (2010) 'General Formal Ontology (GFO): A foundational ontology for conceptual modelling', in *Theory and applications of ontology: computer applications.* Springer, pp. 297-345.

Ireland, C., Bowers, D., Newton, M. and Waugh, K. (2009) 'A classification of object-relational impedance mismatch', *Advances in Databases, Knowledge, and*

Data Applications, 2009. DBKDA'09. First International Conference on. IEEE, 36-43.

Keet, M. (2011) 'The use of foundational ontologies in ontology development: an empirical assessment', in *The Semantic Web: Research and Applications.* Springer, pp. 321-335.

Kent, W. (1978) *Data and reality : basic assumptions in data processing reconsidered.* Amsterdam ; Oxford: North-Holland Publishing Co.

Lowe, E.J. (1998) 'Ontology.', in Hondreich, T. (ed.) *The Oxford Companion to Philosophy.* New York: Oxford University Press, pp. 634.

Lycett, M. (2013) "Datafication': Making sense of (big) data in a complex world', .

Partridge, C. (2002) 'The role of ontology in integrating semantically heterogeneous databases', *Rapport technique,* 5(02).

Partridge, C., Mitchell, A. and de Cesare, S. (2013) 'Guidelines for developing Ontological Architecures in Modelling and Simulation', in Tolk, A. (ed.) *Ontology, Epistemology, and Teleology for Modeling and Simulation.* Berlin Heidelberg: Springer, pp. 27-57.

Office of National Statistics (2016), Postcode Data Product. Available from: < http://www.ons.gov.uk/ons/guide-method/geography/products/postcode-directories/-nspp-/index.html>. [16 February 2016].

Office of National Statistics (2016), Standard Industrial Classification System 2007. Available from: < http://www.ons.gov.uk/ons/guide-method/classifications/current-standard-classifications/standard-industrial-classification/index.html>. [16 February 2016].

Partridge, C. (2005) *Business objects.* 2nd edn. Oxford: Butterworth Heinemann.

Quine, W.V. (1952) *Methods of logic.* Routledge and Kegan Paul.

Saltor, F., Castellanos, M. and Garcia-Solaco, M. (1991) 'Suitability of Data models As Canonical Models for Federated Databases', *SIGMOD Rec.,* 20(4), pp. 44-48.

Sheth, A.P. and Larson, J.A. (1990) 'Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases', *ACM Comput.Surv.,* 22(3), pp. 183-236.

Sider, T. (2003) *Four-dimensionalism: An ontology of persistence and time.* Oxford.

Strawson, P. F. "Identifying reference and truth□values", Theoria, 30(2), 1964 pp. 96-118.

Vicknair, C., Macias, M., Zhao, Z., Nan, X., Chen, Y. and Wilkins, D. (2010) 'A comparison of a graph database and a relational database: a data provenance perspective', *Proceedings of the 48th annual Southeast regional conference.* ACM, 42.

Visser, P.R., Jones, D.M., Bench-Capon, T. and Shave, M. (1997) 'An analysis of ontology mismatches; heterogeneity versus interoperability', *AAAI 1997 Spring Symposium on Ontological Engineering, Stanford CA., USA.* , 164-172.

Webber, J. (2012) 'A programmatic introduction to Neo4j', *Proceedings of the 3rd annual conference on Systems, programming, and applications: software for humanity.* ACM, 217-218.

Zikopoulos, P. and Eaton, C. (2011) *Understanding big data: Analytics for enterprise class hadoop and streaming data.* McGraw-Hill Osborne Media.

Towards a Synthetic Data Generator for Matching Decision Trees

Taoxin Peng and Florian Hanke
School of Computing, Edinburgh Napier University, 10 Colinton Road, Edinburgh, EH10 5DT, U.K.
{t.peng, f.hanke}@mapier.ac.uk

Keywords: Synthetic, Data Generator, Data Mining, Decision Trees, Classification, Pattern.

Abstract: It is popular to use real-world data to evaluate or teach data mining techniques. However, there are some disadvantages to use real-world data for such purposes. Firstly, real-world data in most domains is difficult to obtain for several reasons, such as budget, technical or ethical. Secondly, the use of many of the real-world data is restricted or in the case of data mining, those data sets do either not contain specific patterns that are easy to mine for teaching purposes or the data needs special preparation and the algorithm needs very specific settings in order to find patterns in it. The solution to this could be the generation of synthetic, "meaningful data" (data with intrinsic patterns). This paper presents a framework for such a data generator, which is able to generate datasets with intrinsic patterns, such as decision trees. A preliminary run of the prototype proves that the generation of such "meaningful data" is possible. Also the proposed approach could be extended to a further development for generating synthetic data with other intrinsic patterns.

1 INTRODUCTION

In our modern society in the internet age, collections of data and even more important making use of existing available data gain more and more importance. Especially in the domain of teaching data mining or data mining research, investigators often come across some main problems. Firstly, in order to research or teach a certain problem, most of the techniques and methods in this domain rely on having relevant, big collections of data. It is very common to use real-world data for such purposes. However, real-world data in most domains is difficult to obtain for several reasons, such as budget, technical or ethical (Rachkovskij and Kussul, 1998). Secondly, the use of many of the real-world data is restricted or in the case of data mining, those data sets do either not contain specific patterns that are easy to mine for teaching purposes or the data needs special preparation and the algorithm needs very specific settings in order to find patterns in it. For example, it is also very likely that real data may contain sensible data (be it personal or confidential) which makes it necessary to hide or obscure those parts, resulting in a huge effort to carry out this task because of the sheer size of these data collections. The third problem is that in case of teaching data mining techniques, learners may encounter the same "standard datasets" (e.g. the IRIS dataset or the Cleveland Heart Disease dataset) multiple times during their studies and mining them becomes "less exciting" . This can lower their motivation and as a consequence their learning success.

A solution to these problems could be using synthetic generated data with intrinsic patterns. There are a number of approaches and techniques that have been developed for generating synthetic data (Coyle *et al.*, 2013, Frasch *et al.*, 2011, van der Walt and Bernard, 2007, Sanchez-Monedero et al., 2013, Jeske *et al.*, 2005, Lin *et al.*, 2006, and Pei and Zaiane, 2006). However, since each of the previous research was either focused on a particular category, such as clustering, or using some special techniques, there are still spaces for further research. There is also a survey paper that provides current development about general test data generation tools (Galler and Aichernig, 2014).

This paper presents a novel approach to a synthetic data generator for matching data mining patterns, such as decision trees, by developing a novel decision tree pattern generating algorithm. A preliminary run of the prototype proves that the generation of such big size of "meaningful data" is possible. Also the proposed approach could be extended to a further development for generating synthetic data with other intrinsic patterns.

The rest of this paper is structured as follows. Related works are described in next section. The main contribution of this paper is presented in section 3,

Peng, T. and Hanke, F.
Towards a Synthetic Data Generator for Matching Decision Trees.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 135-141
ISBN: 978-989-758-187-8

which introduces the novel approach, the architecture, the algorithm, the design and implementation of the generator. The testing and evaluation are discussed in section 4. Finally, this paper is concluded and future work pointed out in section 5.

2 RELATED WORK

Sanchez-Monedero *et al* (2013) proposed a framework for synthetic data generation, by adopting a n-spheres based approach. The method allows variables such as position, width and overlapping of data distributions in the n–dimensional space can be controlled by considering their n-spheres. However, this approach only focuses on cases dealing with topics specially in the context of ordinal classifications.

Coyle *et al* (2014) presented a method for estimating data clusters at operating conditions where data has been collected to estimate data at other operating conditions, enabling classification. This can be used in machine learning algorithms when real data cannot be collected. This method uses the earlier mean interpolation along with a method of interpolating all of the matrices comprising the singular value decomposition (SVD) of the covariance matrix to perform data cluster interpolation, based on a methodology termed as Singular Value Decomposition Interpolation (SVDI). It is claimed that the method can be used to yield intuitive data cluster estimates with acceptable distribution, orientation and location in the feature space. However, as authors admitted the method "assumes a uni-model distribution, which may or may not true for classification and regression problems".

Motivated by research work on data characteristics (van der Wlat and Bernard, 2007, Wolpert snd Macready, 1997), Frasch *et al* (2011) proposed a method for generating synthetic data with controlled statistical data characteristics, like means, covariance, intrinsic dimensionality and the Bayes errors. It is claimed that synthetic data generator which can control the statistic properties are important tools for experimental inquiries performed in context of machine learning and pattern recognition. The proposed data generator is suitable for modelling simple problems with fully known statistical characteristics.

Pei and Zaiane (2006) developed a distribution-based and transformation-based approach to synthetic data generation for clustering and outlier analysis. There are a set of parameters that are considered as user's requirements, such as the number of points, the number of clusters, the size, shapes and locations, and the density level of either cluster data or noise/outliers in a dataset. The generator can handle two-dimensional data. However, it was claimed that based on the heuristic devised, the system could be extended to handle three or higher dimensional data.

Jeske *et al*. (2005) proposed an architecture for an information discovery analysis system data and scenario generator that generates synthetic datasets on a to-be-decided semantic graph. Based on this architecture, Lin *et al*. (2006) developed a prototype of this system, which is capable of generating synthetic data for a particular scenario, such as credit card transactions.

The work probably most closely related to the one proposed in this paper is the one by Eno and Thompson (2008). The authors proposed an approach toward determining whether patters found by data mining models could be used and reverse map them back into synthetic data sets of any size that would exhibit the same patterns, by developing an algorithm to map and reverse a decision tree. Their approach was based on two technologies: Predictive Model Markup Language (PMML) and Synthetic Data Definition Language (SDDL). The algorithm would scan a decision tree stored as PMML to create an SDDL file that described the data to be generated. It was claimed that their method confirmed the viability of using data mining models and inverse mapping to inject realistic patterns into synthetic data sets. However, their work is limited to the two techniques used.

3 THE APPROACH

This section describes the proposed framework, including the architecture, the pattern generating algorithm, the design and implementation of the approach.

3.1 Architecture

Figure 1 illustrates the relationship of all modules in the framework. These modules can be implemented to run in separate threads or even on separate systems to create a distributed system which would optimise the performance of the whole application. The architecture is a modified version of the one proposed by Houkjær *et al*. (2006).

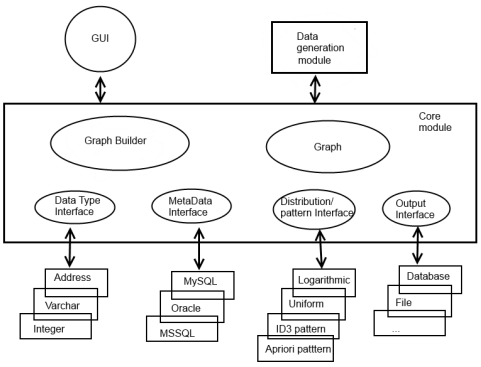


Figure 1: The Architecture.

Main components in this architecture are described as below:

- **GUI:** This package contains all the classes necessary for the graphical user interface. The GUI classes enable the user
  - to set parameters and inputs;
  - to choose and set up the connection to the database;
  - to view the meta data connected to the tables in the database;
  - to choose from a list of available data generation algorithms/methods;
  - to set the desired output formats.
- **Data Generation Module:** This package contains the classes needed to generate data, e.g. different number generators (such as zero bitmap number generators, shuffle number generators or specialised number generators), classes that can produce addresses or names and so on;
- **The Core Module:** This package contains three sub packages:
  - **Graph Builder:** This sub package contains all classes necessary to generate a directed graph which represents the database/table structures retrieved from the database through the Metadata Interface;
  - **Graph:** The graph sub package holds a representation of the database in memory. This is necessary in order to generate consistent data that fulfils constraints as well as intra- and inter-table relations;
  - **Interfaces:** This sub package contains the interfaces and their class implementations which are used by the graph, graph builder and data generation module classes and provide the different ways of input (different DBMS, e.g. MySQL, Oracle, etc.; inputs for name/address generation), output (e.g. into flat files) and the interfaces used for the different data generation algorithms or number distributions. One of the most important interfaces in this design is the pattern interface. This interface can be used together with the new approach to pattern generation in data to form a really unique data generator.

3.2 A Decision Tree Algorithm: ID3

This new approach employs the idea of "Backwards Engineering": an existing well established classification algorithm (in this case ID3) is used as the basis to discover the patterns; then an algorithm is developed that produces data in way such that this basis algorithm is able to discover a structure in the data.

In this framework, the well-known ID3 algorithm, originated by Quinlan (1979, 1986) was used following the description of Berthold *et al* (2010):

Algorithm BuildDecisionTree($\mathcal{D}, \mathcal{A}$)

input: training data $\mathcal{D}$, set $\mathcal{A}$ of available attributes
output: a decision tree matching $\mathcal{D}$, using all or a subset of $\mathcal{A}$

```
1   if all elements in D belong to one class
2     return node with corresponding class label
3   elseif A = ∅
4     return node with majority class label in D
5   else
6     select attribute A ∈ A which best classifies D
7     create new node holding decision attribute A
8     for each split v_A of A
9       add new branch below with corresponding test for this split
10      create D(v_A) ⊂ D for which split condition holds
11      if D(v_A) = ∅
12        return node with majority class label in D
13      else
14        add subtree returned by calling
              BuildDecisionTree(D(v_A), A\{A})
15      endif
16    endfor
17    return node.
18  endif
```

Figure 2: The ID3 algorithm as described by Berthold *et al.* (Berthold et al., 2010, p. 211).

Figure 2 shows a general algorithm to build decision trees. ID3 in particular uses a concept called the Shannon Entropy *H*:

$$H_{\mathcal{D}}(\mathcal{C}) = -\sum_{k \in \mathrm{dom}(\mathcal{C})} \frac{|\mathcal{D}_{\mathcal{C}=k}|}{|\mathcal{D}|} \log \frac{|\mathcal{D}_{\mathcal{C}=k}|}{|\mathcal{D}|}$$

Here, $\mathcal{D}$ indicates the training data set, $\mathcal{C}$ the target (class) attribute, i.e. the attribute towards which the entropy is calculated, and $\mathcal{A}$ the set of attributes. The entropy ranges from 0 to 1 and reaches the maximal value of 1 for the case of two classes and an even 50:50 distribution of patterns of those classes. On the other hand, an entropy value of 0 would tell us that only one of these classes would exist in the given subset of data. The entropy *H* therefor provides us with a measure of the diversity of a given data set.

The ID3 algorithm tries to reach the leaves of a decision tree (i.e. nodes that only hold a single class of attributes) as fast as possible, meaning that the entropy of each subset of data after the split of the values should have the least possible entropy. Therefore, another measurement is needed, called the "Information Gain":

$$I_{\mathcal{D}}(\mathcal{C}, A) = H_{\mathcal{D}}(\mathcal{C}) - H_{\mathcal{D}}(\mathcal{C}, A)$$

Where

$$H_{\mathcal{D}}(\mathcal{C}, A) = \sum_{a \in \mathrm{dom}(A)} \frac{|\mathcal{D}_{A=a}|}{|\mathcal{D}|} H_{\mathcal{D}_{A=a}}(\mathcal{C})$$

and $\mathcal{D}_{A=a}$ indicates the subset of $\mathcal{D}$ for which attribute A has value a. $H_{\mathcal{D}}(\mathcal{C}, A)$ denotes the entropy that is left in the subsets of the original data after they have been split according to their values of A.

This Information Gain makes it possible to split the classes in $\mathcal{D}$ into subsets with each having the least possible remaining entropy within. Using this Information Gain as measurement in the split condition for the Class attribute of the algorithm outlined in figure 2, the ID3 algorithm is complete.

3.3 The Algorithm

With the ID3 algorithm and its underlying concepts defined, the pattern generating algorithm can be described.

The requirements for this algorithm are a classification decision tree with a table in a database having at least columns for each of the attributes that are present in the nodes of the tree and the Class attribute. In contrast to the ID3 algorithm that will later be used to find the same tree again, the proposed pattern generating algorithm does not start from the root of the tree, working its way "downwards" over nodes with the highest Information Gain to the leave nodes, but it starts from the leave nodes in an "upward" way.

The basic idea of the algorithm can be described as follows. The leave nodes L have to be the nodes with the *least* Information Gain of the whole data set. This can be ensured by *maximally* distributing the values of the Class attribute C on this level (this will of course result in a very inaccurate classification tree; in the implementation different distribution levels can be used to make it more accurate). To do this, a minimum number of entries in the database table has to be specified; according to this number, the table is then populated with maximal distribution in C (which means all possible value c in C appears with the same frequency), leaving all other columns blank with the exception of the values in L (noted as l in future). These are then chosen such that each combination of l and c appears equally.

Now, when c is maximally distributed among l, the entropy of L in respect to C is 1 and since the Information Gain can never be negative and the range of entropy is between 0 and 1, the Information Gain for L is 0 and ID3 will use L as the leave nodes when the other attributes have a higher Information Gain.

For the next level of nodes N_1 in the given classification tree, all that has to be done is to make sure the entropy for this level is a little lower than the previous one, the easiest way to ensure this is to add one more combination of a specific value of c and a specific value n_1 of N_1; the rest of the combinations should stay maximally distributed (again, in the implementation this "step width" can be set to different values). To achieve this, a number of rows depending on the number of different values of c rows have to be added. Only the distribution among the combinations of n_1 and c must be altered, not the distribution of combinations of l and c. This will result in an entropy value slightly lower than 1 for the attribute N_1 in respect to C thus this attribute N_1 will be used in the node level just above the leaves.

For the next node level N_2 (again having the different values n_2) in the classification tree, not only one specific combination of n_2 and c has to be added but two, therefore two times the number of values c of rows have to be added to keep the combinations of c and l maximally distributed and the combinations of c and n_1 slightly less distributed.

This means again the entropy $H(N_2|C) < H(N_1|C) < H(L|C)$ and in that way, L will be found as leaves by ID3, N_1 as the node level above the leaves, N_2 as the node level above N_1 and if this procedure is repeated until the root of the classification tree. The database table will grow with each step. But the entropy of each attribute higher to the top of the input classification tree will be lower than the entropy to the attributes closer to the leave nodes, which means their Information Gain is higher. Thus ID3 will place them into the right position.

3.4 Implementation

This section describes the implementation of the algorithm outlined above.

3.4.1 Overview of the Implementation

Figure 3 shows the complete class diagram of the prototype. The implementation of the pattern generating algorithm is split among three main classes:

- the "Tree" class provides the framework for the classification tree data structure required for the algorithm;
- the "Node" class provides all the methods and functions necessary to traverse the tree, get certain nodes and update the entropy values accordingly;
- the "TestMain" class makes the use of this data structure, sets the entropy values of the different Node levels and finally also deals with the data generation;

In addition to the three main classes, there are two helper classes, "BSGTree" and "BSGTreeBean":

- "BSGTree" class defines and builds the tree data structure utilising the "Tree" and "Node" classes;
- the "BSGTreeBean" class is a simple Java Bean with private members and Getters and Setters for them. It is used by the "TestMain" class in order to generate the data.

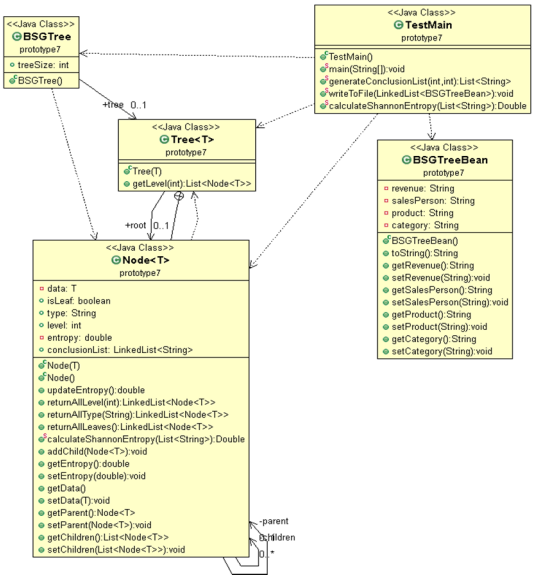


Figure 3: Class diagram.

3.4.2 The Implementation

The prototype only includes the implementation of the pattern generating algorithm, which can be described as the following steps:

- first of all, conclusion lists containing the values of the class attribute are generated with different entropy values;
- then, each of these lists is used to set the conclusion lists of the nodes in one level. Hence, these lists define the starting entropy for level 0 and the "step width" as described in the "description of the algorithm" section;
- the next step is the generation of the predefined tree data structure followed by getting node lists for the different types and levels. With these node lists, each level can be populated with conclusion lists with increasing entropy values;
- further, after the above are all done, each row of data has to be generated. As stated before, each entry in the conclusion lists of the leaf nodes represents a complete data set to be generated. Consequently in order to generate the data rows, all of the leaf nodes can be retrieved by the tree and then their conclusion lists can be looped through; the parent nodes of the leaf nodes recursively contain the values of other attributes. Of course, some attributes might be missing in the chain from a leaf node to the root node. These missing values are replaced by a placeholder value and handled later. All of these row data is collected in a list of beans of the corresponding tree.
- finally, the placeholder values have to be replaced with real attribute values. It is of high importance that the entropy values for the different attributes are not altered in this step. This could happen easily if the placeholder vales are not replaced carefully.

The generated data then can be exported after optionally shuffling the resulting rows.

4 TESTING AND EVALUATION

4.1 Testing

The proposed pattern generator was tested by arbitrarily generating three datasets with three different types of classification trees constructed in, and then finding the patterns in each of the dataset by the J48 classification algorithm of WEKA.

Testing results are shown in figures 4, 5 and 6.

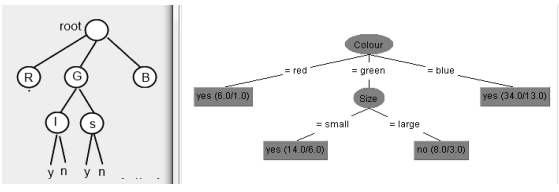


Figure 4: Left: Test tree 1. Right: Tree found by WEKA J48.

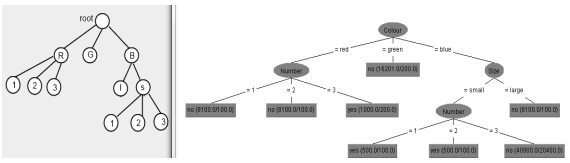


Figure 5: Left: Test tree 2. Right: Tree found by WEKA J48.

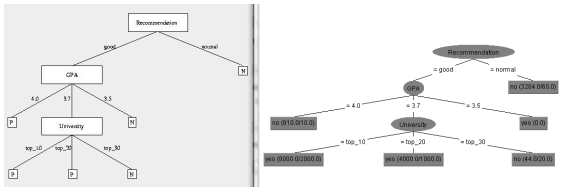


Figure 6: Left: Test tree 3. Right: Tree found by WEKA J48.

Figure 4 shows a simple tree with only 6 notes constructed in a generated dataset at the left hand side and the tree found by the J48 algorithm in Weka at the right hand side. Figure 5 and 6 shows the similar practice with a little bit more complicated tree structures in generated datasets. In all of the testing cases, the designed tree structures were found successfully in the generated datasets, respectively.

4.2 Evaluation

The test cases show that it is definitely possible to generate data that matches a data mining pattern. In some cases, the entropy step width had to be altered or additional "hidden nodes" had to be introduced to the tree in order to make some splits. But this is most likely due to the fact that the pattern generator algorithm's implementation is not technically mature yet and can be improved in further versions. Furthermore, a module should be developed that reads trees as XML files (or similar) and generates the tree structure necessary to generate the data automatically. This would greatly increase the versatility of the synthetic data generator.

In summary, the testing results prove that the proposed synthetic data generator is able to generate datasets with intrinsic patterns, such as decision trees. Additionally, the performance of the data generator was surprisingly good. It was possible to create almost a million rows in a few seconds with a laptop with basic specifications.

5 CONCLUSIONS AND FUTURE WORK

In this paper, a novel approach for developing a synthetic data generator for matching decision trees has been proposed. A prototype of such a generator has been implemented. The results of the test run prove that a large dataset with patterns like decision trees can be generated automatically within seconds.

While the prototype meets all requirements set out within the aims of the project, the work introduces a number of further investigations, including: a) to add more classification algorithms into the generator; b) to add more algorithms into the generator, which allow patterns of association rules, clustering and repression to be created; c) to develop a comprehensive, user-friendly interface, which allows users to select algorithms from different categories, define the number of attributes, and other parameters. The successful outcome of such future work would result in a comprehensive synthetic data generator, which is able to generate big datasets with patterns for data mining research and training.

REFERENCES

Berthold, M., Borgelt, C., Höppner, F., & Klawonn, F. 2010. *Guide to intelligent data analysis: How to intelligently make sense of real data.* Springer-Verlag London.

Coyle, E., Roberts, R., Collins, E., and Barbu, A. 2014. Synthetic Data Generation for Classification via Uni-Modal Cluster Interpolation. *Auto Robot* 37:27 - 45.

Eno, J. and Thompson, C., 2008. Generating Synthetic Data to Match Data Mining Patterns. *IEEE Intenet Computing*, Vol. 12, No. 3 pp. 78 – 82.

Frasch, J. V., Lodwich, A., Shafait, F. and M. Breuel, T. M., 2011. A Bayes-true data generator for evaluation of supervised and unsupervised learning Methods. *Pattern Recognition Letters* 32.11, pp. 1523–1531.

Galler, S. J. and Aichernig, B. K. 2014. An Evalaution of White- and Grey-box Testing Tools for C#, C++, Eiffel, and Java, *Int J Softw Tools Technol Transfer* 16: pp. 727 -751.

Houkjær, K., Torp, K., and Wind, R. 2006. Simple and Realistic Data Generation. *Proceedings of the 32nd international conference on very large data bases (VLDB '06), pp. 1243-1246*

Jeske, D. R., Samadi, B., Lin, P. J., Ye, L., Cox, S., Xiao, R., Younglove, T., Ly, M., Holt, D., and Rich, R., 2005. Generation of Synthetic Data Sets for Evaluating the Accuracy of Knowledge Discovery Systems. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in*

Data Mining. ACM, New York, NY, USA. pp. 756 – 762.

Lin, P., Samadi, B., Cipolone, A., Jeske, D., Cox, S., Rendon, C., Holt, D. and Xiao, R., 2006. Development of a Synthetic Data Set Generator for Building and Testing Information Discovery Systems. In *Proceedings of the Third International Conference on Information Technology: New Generations*. IEEE, pp. 707 - 712

Pei, Y. and Zaiane, O., 2006. A Synthetic Data Generator for Clustering and Outlier Analysis. Technical Report, University of Alberta, Canada.

Quinlan, J. R. 1979. Discovering Rules by Induction from Large Collections of Examples. In D. Michie (Ed.), *Expert Systems in the Micro Electronic Age*. Edinburgh University Press.

Quinlan, J. R. 1986. Induction of Decision Trees, *Machine Learning* 1: 81-106.

Rachkovskij, D. A. and Kussul, E. M., 1998. Datagen: A Generator of Datasets for Evaluation of Classification Algorithms. *Pattern Recognition Letters* 19 (7), 537-544.

Sánchez-Monedero, J., Gutiérrez, P. A., Pérez-Ortiz, M. and Hervás- Martínez, C. 2013. An n-Spheres Based Synthetic Data Generator for Supervised Classification. *Advances in Computational Intelligence*. Ed. by Rojas, I., Joya, G. and Gabestany, J. *Lecture Notes in Computer Science 7902*. Springer Berlin Heidelberg, pp. 613–621.

van der Walt, C. and Barnard, E. 2007. Data Characteristics That Determine Classifier Performance. *SAIEE Africa Research Journal, Vol 98(3), pp 87-93*.

Document-oriented Models for Data Warehouses
NoSQL Document-oriented for Data Warehouses

Max Chevalier[1], Mohammed El Malki[1,2], Arlind Kopliku[1], Olivier Teste[1] and Ronan Tournier[1]
[1]*Université de Toulouse, IRIT (UMR 5505), Toulouse, France*
[2]*Capgemini, Toulouse, France*
{Max.Chevalier, Mohammed.ElMalki, Arlind.Kopliku, Olivier.Teste, Ronan.Tournier}@irit.fr,
Mohammed.El-Malki@capgemini.com

Keywords: NoSQL, Document-oriented, Data Warehouse, Multidimensional Data Model, Star Schema.

Abstract: There is an increasing interest in NoSQL (Not Only SQL) systems developed in the area of Big Data as candidates for implementing multidimensional data warehouses due to the capabilities of data structuration/storage they offer. In this paper, we study implementation and modeling issues for data warehousing with document-oriented systems, a class of NoSQL systems. We study four different mappings of the multidimensional conceptual model to document data models. We focus on formalization and cross-model comparison. Experiments go through important features of data warehouses including data loading, OLAP cuboid computation and querying. Document-oriented systems are also compared to relational systems.

1 INTRODUCTION

In the area of Big Data, NoSQL systems have attracted interest as mean for implementing multidimensional data warehouses (Chevalier et al, 2015a), (Chevalier et al, 2015b), (Mior, 2014), (Dede et al, 2013), (Schindler, 2012). The proposed approaches mainly rely on two specific classes of NoSQL systems, namely document-oriented systems (Chevalier et al, 2015a) and column oriented systems (Chevalier et al, 2015b), (Dede et al, 2013). In this paper, we study further document-oriented systems in the context of data warehousing.

In contrast to Relational Database Management Systems (RDBMS), document-oriented systems,and many other NoSQL systems, are famous for horizontal scaling, elasticity, data availability, and schema flexibility. They can accommodate heterogeneous data (not all conforming to one data model); they provide richer structures (arrays, nesting…) and they offer different options for data processing including map-reduce and aggregation pipelines. In these settings, it becomes interesting to investigate for new opportunities for data warehousing. On one hand, we can exploit scalability and flexibility for large-scale deployment. On the other hand, we can accommodate heterogeneous data and consider mapping to new data models. In this setting, document-oriented systems become natural candidates for implementing data warehouses.

In this paper, we consider four possible mappings of the multidimensional conceptual model into document logical models. This includes simple models that are analogous to relational database models using normalization and denormalization. We also consider models that use specific features of the document-oriented system such as nesting and schema flexibility. We instantiate a data warehouse using each of the models and we compare each instantiation with each other on different axes including: data loading, querying, and OLAP cuboid computation.

2 RELATED WORK

Multidimensional databases are mostly implemented using RDBMS technologies (Chaudhuri et al, 1997), (Kimball, 2013). Considerable research has focused on the translation of data warehousing concepts into relational logical level (Bosworth et al, 1995), (Colliat et al, 1996), (called R-OLAP). Mapping rules are used to convert structures of the conceptual level (facts, dimensions and hierarchies) into a logical model based on relations (Ravat, et al, 2006).

Chevalier, M., Malki, M., Kopliku, A., Teste, O. and Tournier, R.
Document-oriented Models for Data Warehouses - NoSQL Document-oriented for Data Warehouses.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 142-149
ISBN: 978-989-758-187-8

There is an increasing attention towards the implementation of data warehouses with NoSQL systems (Chevalier et al, 2015a), (Zhao et al, 2014), (Dehdouh et al, 2014), (Cuzzocrea et al, 2013). In (Zhao et al, 2014), the authors implement a data warehouse into a column-oriented store (HBase). They show how to instantiate efficiently OLAP cuboids with MapReduce-like functions. In (Floratou et al, 2012), the authors compare a column-oriented system (Hive on Hadoop) with a distributed version of a relational system (SQL server PDW) on OLAP queries.

Document-oriented systems offer particular data structures such as nested sub-documents and arrays. These features are also met in object-oriented and XML like systems. However, none of the above has met success as RDBMS for implementing data warehouses and in particular for implementing OLAP cuboids as we do is this paper. In (Kanade et al, 2014), different document logical models are compared to each other: data denormalization, normalized data; and models that use nesting. However, this study is in a "non-OLAP" setting.

In our previous work (Chevalier et al, 2015a), (Chevalier et al, 2015b) we have studied 3 column-oriented models and 3-document-oriented models for multidimensional data warehouses. We have focused on direct translation of the multidimensional model to NoSQL logical models. However, we have considered simple models (models with few document-oriented specific features) and the experiments were at an early stage. In this paper, we focus on more powerful models and our experiments cover most of data warehouse issues.

3 DOCUMENT DATA MODEL FOR DATA WAREHOUSES

We distinguish three abstraction levels: *conceptual model* (Golfarelli et al, 1998), (Annoni, et al, 2006) that is independent of technologies, *logical model* that corresponds to one specific technology but software independent, *physical model* that corresponds to one specific software. The multidimensional schema is the reference conceptual model for data warehousing. We will map this model to document-oriented data models.

3.1 Multidimensional Conceptual Model

Definition 1. A *multidimensional schema*, namely E, is defined by $(F^E, D^E, Star^E)$ where: $F^E = \{F_1, ..., F_n\}$ is a finite set of facts, $D^E = \{D_1, ..., D_m\}$ is a finite set of dimensions, and $Star^E$: $F^E \rightarrow 2^{D^E}$ is a function that associates facts of F^E to sets of dimensions along which it can be analyzed (2^{D^E} is the *power set* of D^E).

Definition 2. A *dimension*, denoted $D_i \in D^E$ (abusively noted as D), is defined by (N^D, A^D, H^D) where: N^D is the name of the dimension; $A^D = \{a_1^D, \dots, a_u^D\} \cup \{id^D, \dots, All^D\}$ is a set of dimension attributes; and $H^D = \{H_1^D, \dots H_v^D\}$ is a set of hierarchies. A hierarchy can be as simple as the example *{"day, month, year"}*.

Definition 3. A *fact*, $F \in F^E$, is defined by (N^F, M^F) where: N^F is the name of the fact, and $M^F = \{m_1^F, \dots, m_v^F\}$ is a set of measures. Typically, we apply aggregation functions on measures. A combination of dimensions represents the analysis axis, while the measures and their aggregations represent the analysis values.

3.2 Document-oriented Logical Model

Here, we provide key definitions and notation we will use to formalize documents. Documents are grouped in collections. We refer to such a document as *C(id)*.

Definition 4. A *document* corresponds to a set of key-values. A unique key identifies every document; we call it identifier. Keys define the structure of the document; they act as meta-data. Each value can be an atomic value (number, string, date…) or a sub-document or array. Documents within documents are called sub-documents or nested documents.

Definition 5. The *document structure/schema* corresponds to a generic document without atomic values i.e. only keys.

We use the colon symbol "**:**" to separate keys from values, "**[]**" to denote arrays, **"{}"** to denote documents and a comma **","** to separate key-value pairs from each other.

With the above notation, we can provide an example of a document instance. It belongs to the "Persons" collection, it has 30001 as identifier and it contains keys such as "name", "addresses", "phone". The addresses value corresponds to an array and the phone value corresponds to a sub-document.

```
   Persons(30001):
{name:"John Smith",
 addresses:
 [{city:"London", country:"UK"},
  {city:"Paris", country:"France"}],
 phone:
 {prefix:"0033", number:"61234567"}}
```

The above document has a document schema:

```
 {name, addresses: [{city, country}],
phone: {prefix, number}}
```

Another way to represent a document is through all the paths within the document that reach the atomic values. A path *p* of a document instance with identifier `id` is described as `p=C(id):k1:k2:…kn:a` where `k1, k2,… kn:a` are keys within the same path ending at an atomic value `a`.

In a same collection it is possible to have documents with different structures: the schema is specific at the document level. We define the collection model as the union of all schemas of all documents. A collection *C* that accepts two sub-models S_1 and S_2, can be written as $S^C=\{S_1, S_2\}$. This formalism will be enough for our purposes.

3.3 Document-oriented Models for Data Warehousing

In this section, we present document models that we will use to map the multidimensional data model. We refer here to the multidimensional conceptual model as described in section 3 and we describe and illustrate four logical data models. Each time we describe the model for a fact *F* (with name N^F) and its dimensions $D \in Star^E(F)$ (each dimension has a name N^D).

We will illustrate each model with a simple example. We consider the fact "LineOrder" and only one dimension "Customer". For "LineOrder", we have three measures "l_quantity", "l_shipmode" and "l_price". For "Customer", we have three attributes "c_name", "c_city" and "c_nation_name".

The chosen models are diverse each one with strengths and weaknesses. They are also useful to illustrate the modeling issues in document-oriented systems. Models **M0** and **M2** are equivalent to data denormalization and normalization in RDBMS. Model **M1** is similar to **M0**, but it adds some more structure (meta-data) to documents. This model is interesting to see if extra meta data is penalizing (in terms of memory usage, query execution, etc.). Model **M3** is similar to **M2**, but everything is stored in one collection. **M3** exploits schema flexibility i.e. it stores in one collection documents of different schema.

Each model is defined, formalized and illustrated below:

Model M0, Flat: It corresponds to a denormalized flat model. Every fact from *F* is stored in a collection C^F with all attributes of its dimensions $Star^E(F)$. It corresponds to denormalized data (in RDBMS). Documents are flat (no nesting), all attributes are at the same level. The schema S^F of the collection C^F is:

$$S_F = \{\mathrm{id}, m_1, m_2, \dots m_{|M^F|}, a_1^{D_1}, a_2^{D_1}, \dots a_{|A^{D_1}|}^{D_1}, a_1^{D_2}, a_2^{D_2} \dots a_{|A^{D_2}|}^{D_2}, \dots\}$$

e.g.

```
{id:1,
 l_quantity:4,
 l_shipmode:"mail",
 l_price:400.0,
 c_name:"John",
 c_city:"Rome",
 c_nation_name:"Italy"}
```

Model M1, Deco: It corresponds to a denormalized model with more structure (meta-data). It is similar to M0, because every fact *F* is stored in a collection C^F with all attributes of its dimensions $Star^E(F)$. In each document, we group measures together in a sub-document with key N^F. Attributes of one dimension are also grouped together in a sub-document with key N^D. This model is simple, but it illustrates the existence of non-flat documents. The schema S^F of the C^F is:

$$S_F = \left\{id^F, \mathrm{N}^F: \left\{m_1, m_2, .. m_{|M^F|}\right\}, N^{D_1}: \{a_1^{D_1}, a_2^{D_1}, \dots a_{|A^{D_1}|}^{D_1}\right\}, N^{D_2}: \{a_1^{D_2}, a_2^{D_2}, \dots a_{|A^{D_1}|}^{D_2}\}, \dots\}$$

e.g.

```
{id:1,
 LineOrder:
  {l_quantity:4,
   l_shipmode:"mail",
   l_price:400.0},
 Customer:
  {c_name:"John",
   c_city:"Rome",
   c_nation_name:"Italy"}}
```

Model M2, Shattered: It corresponds to a data model where fact records are stored separately from dimension records to avoid redundancy, equivalent to normalization. The fact *F* is stored in a collection C^F and each dimension $D \in Star^E(F)$ is stored in a collection C^D**.** The fact documents contain foreign keys towards the dimension collections. The schema S^F of C^F and the schema S^D of a dimension collection C^D are as follows:

$$S_F = \{id_F, m_1, m_2, \dots m_{|M^F|}, id_{D_1}, id_{D_2}, \dots\}$$

$$S_D = \{id_D, a_1^{\mathrm{D}}, a_2^{\mathrm{D}}, \dots a_{|A^D|}^{\mathrm{D}}\}$$

e.g.

```
{id:1,
 l_quantity:4,
 l_shipmode:"mail",
 l_price:400.0,
 c_id:4} ∈ C^F
{id:4,
 c_name:"John",
 c_city:"Rome",
 c_nation_name:"Italy"} ∈ C^Customer
```

Model M3, Hybrid: It corresponds to a hybrid model where we store documents of different schema in one collection. We store everything in one collection, say C^F. We store the fact entries with a schema S^F. Dimensions are stored within the same collection, but each with its complete schema S^D.

We need to keep references from fact entries towards the corresponding dimension entries. This model is similar to M2, at the difference of storing everything in one collection.

This model is interesting, because if we use indexes properly, we can access quickly the dimension attributes and all corresponding facts e.g. with an index on *c_custkey*, we access quickly all sales of a given customer.

The schemas S^F and S^D are:

$$S_F = \{id, m_1, m_2, \dots m_{|M^F|}, id^{D_1}, id^{D_2}, \dots\};$$
$$S_D = \{id^D, \quad a_1^D, a_2^D, \dots a_{|A^F|}^D\}$$

e.g.

```
{id:1,
 l_quantity:4,
 l_shipmode:"mail",
 l_extended_price:400.0,
 c_custkey:2,
 c_datekey:3} ∈ C^F
{id:2,
 custkey: 4,
 c_name: "John",
 c_city: "Rome",
 c_nation_name:"Italy",
 c_region_name:"Europe"} ∈ C^F
{id:3,
 date_key:1,
 d_date:10,
 d_month:"January",
 d_year:2014} ∈ C^F
```

In Table 1, we summarize the mapping of the multidimensional model to our logical models. For every dimension attribute or fact measure, we show the corresponding collection and path within a document structure.

Table 1: Mapping of the multidimensional schema to the logical data models.

	$\forall D \in D^O \ \forall a \in A^D$		$\forall m \in M^F$	
	collection	*path*	*collection*	*path*
M0	C^F	*a*	C^F	*m*
M1	C^F	N^D:*a*	C^F	N^F:*m*
M2	C^D	*a*	C^F	*m*
M3	C^F	*a*	C^F	*m*

4 EXPERIMENTS

4.1 Experimental Setup

The experimental setup is briefly introduced and then detailed in the next paragraphs. We generate 4 datasets according to the SSB+, Star schema benchmark (Chevalier et al, 2015c), (Oneil et al, 2009), which is itself a derived from the TPC-H benchmark. TPC-H is a reference benchmark for decision support systems. The benchmark is extended to generate data compatible to our document models (M0, M1, M2, M3). Data is loaded in MongoDB v2.6, a popular document-oriented system. On each dataset, we issue sets of OLAP queries and we compute OLAP cuboids on different combinations of dimensions. Experiments are done in single-node and a distributed 3-nodes cluster setting.

For comparative reasons, we also load two datasets in PostgresSQL v8.4, a popular RDBMS. In this case, dataset data corresponds to a flat model (M0) and a star-like normalized model (M2), that we name respectively R0 and R2. Experiments in PostgreSQL are done in a singlenode setting.

Data. We generate data using an extended version of the Start Schema Benchmark denoted SSB+ (Chevalier et al, 2015c), (Oneil et al, 2009). The benchmark models a simple product retail reality. The SSB+ benchmark models a simple product retail reality. It contains one fact "LineOrder" and 4 dimensions "Customer", "Supplier", "Part" and "Date".

We generate data using an extended version of the Start Schema Benchmark SSB (Oneil et al, 2009) because it is the only data warehousing benchmark that has been adapted to NoSQL systems. The extended version is part of our previous work (Blind3). It makes possible to generates raw data directly as JSON which is the preferable data format for data loading in MongoDB. We use improve scaling factor issues that have been reported. In our experiments we use different scale factors (*sf*) such as *sf=1, sf=10* and *sf=25* in our experiments. In the extended version, the scale factor *sf=1* corresponds to approximately 10^7 records for the LineOrder fact, for *sf=10* we have approximately $10x10^7$ records and so on.

Settings/Hardware/Software. The experiments have been done in two different settings: single-node architecture and a cluster of 3 physical nodes. Each node is a Unix machine (CentOs) with 4 core-i5 CPU, 8GB RAM, 2TB disks, 1Gb/s network. The cluster is composed of 3 nodes, each being a worker node and one node acts also as dispatcher. Each node has a MongoDB v.3.0 running. In MongoDB terminology,

this setup corresponds to 3 shards (one per machine). One machine also acts as configuration server and client.

4.2 Document-oriented Data Warehouses by Model

Data Loading. We report first the observations on data loading. Data with model M0 and M1 occupy about 4 times less space than data with models M2 and M3. For instance, at scale factor *sf=1* (10^7 line order records) we need about 4.2GB for storing models M2 and M3, while we need about 15GB for models M0 and M1. The above observations are explained by the fact that data in M2 or M3 has less redundancy. In M2 and M3 dimension data is repeated just once.

Figure 1 shows data loading times by model and scale factor (*sf=1, sf=10, sf=25*) on a singlenode setting. Loading times are as expected higher for the data models that require more memory (M0 and M1). In Figure 2, we compare loading times for sf=1 on singlenode setting with the distributed setting. We observe data loading is significantly slower in a distributed setting than on a single machine. For instance, model M0 data (*sf=1*) loads for 1306s on a single cluster, while it needs 4246s in a distributed setting. This is mainly due to penalization related to network data transfer. Indeed, MongoDB balances data load i.e. it tries to distribute equally data across all shards implying more network communication.

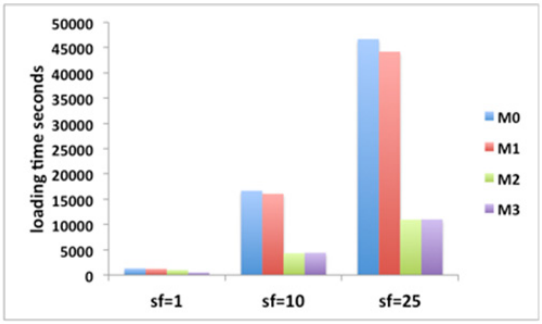


Figure 1: Loading times by data models.

Querying. We test each instantiation (on 4 data models) on 3 sets of OLAP queries (QS1, QS2, QS3). To do so, we use the SSB benchmark query generator that generates 3 query variants per set. The query complexity increases from QS1 to QS3. QS1 queries filter on one dimension and aggregate all data; QS2 queries filter data on 2 dimensions and group data on one dimension; and QS3 queries filter data on 3 dimensions and group data on 2 dimensions.

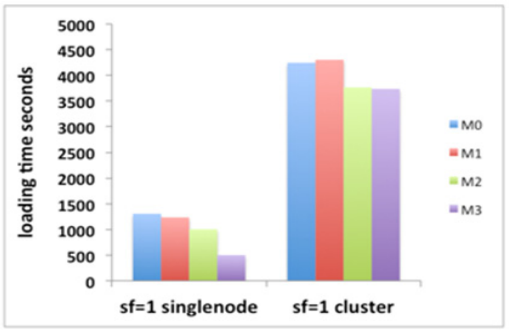


Figure 2: Loading time comparisons on single node and cluster.

In Table 3 and 4, we show query execution times on all query variants with scale factor *sf=1,* all models, in two settings (single node and cluster). For the queries with 3 variants, results are averaged (arithmetic mean). In Table 3, we can compare averaged execution times per query and model in the single node setting. In Table 4, we can compare execution times in the distributed (cluster) setting.

We observe that for some queries some models work better and for others some other models work better. We would have expected queries to run faster on models M0 and M1 because data is in a denormalized fashion (no joins needed). This is surprisingly not the case. Query execution times are comparable across all models and sometimes queries run faster for models M2 and M3. This is partly because we could optimize queries choosing from the MongoDB rich palette: aggregation pipeline, map/reduce, simple queries and procedures. For M2 and M3, we need to join data from more than one document at a time. When we do not write the most efficient MongoDB query and/or when we join all data needed for the query before any filtering, execution times can be significantly higher. Instead we apply filters before joins and then we use the aggregation pipeline , map/reduce functions, simple queries or procedures. We also observed the SSB queries had high selectivity. We could filter most records before needing any join. To test selectivity impact, we tested querying performance on another query Q4 that is obtained by modifying one of the queries from QS1 to be more selective. On this new query set we have about 500000 facts after filtering. We observe that query execution on data with models M0 and M1 is lower about 20-30%. Meanwhile, on data with models M2 and M3 query execution is respectively about 5-15 times slower. This is purely due to the impact of joins that are not supported by document-oriented systems in general.

To fully understand the impact of joins on data with models M2 and M3, we conducted another experiment when we join all data i.e. we basically generate data with model M0 starting from data with model M2 and M3. In the most performant approaches we could produce, we observed 1010 minutes for M2 and 632 minutes for M3 on *sf=1*. This is a huge delay. We can conclude that data joins can be a major limitation for document-oriented system. When joins are poorly supported, data models such as M2 and M3 are not interesting.

In Table 3 and Table 4, we can also compare query execution times in singlenode setting with respect to distributed setting. We observe that query execution times are generally better in a distributed setting. For many queries, execution times improve 2 to 3 times depending on the cases. In a distributed setting, query execution is penalized by network data transfer, but it is improved by parallel computation. When queries are executed on data with models M2 and M3, improvement on the distributed setting is less important (less than 1.5 times).

4.3 OLAP Cuboids with Documents

OLAP Cuboid. It is common in OLAP applications to pre-compute analysis cuboids that aggregate fact measures on different dimension combinations. In our example (SSB dataset), there are 4 dimensions C: Customer, S: Supplier, D: Date and P: Part. In Figure 3, we show all possible dimension combinations. Data can be analyzed on no dimension (all), 1 dimension, 2 dimensions or 3 dimensions or 4 dimensions. Cuboid names are given with dimension initials, e.g. CSP stands for cuboid on Customer, Supplier and Part. In Figure 3, we show for illustration purposes the computation time for a complete lattice in M0. In this case, we compute lower level cuboids from the cuboid just on top to make things faster.

In Table 2 we show the average time needed to compute an OLAP cuboid of *x* dimensions (*x* can be 3, 2, 1, 0, i.e. group on 3 dimensions, 2 dimensions and so on). Cuboids are produced starting from data on any of the models M0, M1, M2, or M3.

Table 2: Average aggregation time per lattice level on single node setting.

	M0	**M1**	**M2**	**M3**
3D	423s	460s	303s	308s
2D	271s	292s	157s	244s
1D	196s	201s	37s	44s
all	185s	191s	37s	27s

We observe that we need less time to compute the OLAP cuboid with M2 and M3. This is because we do not denormalize data, i.e. we group only on foreign keys. If we need cuboids that use other dimension attributes, the computation time is significantly higher.

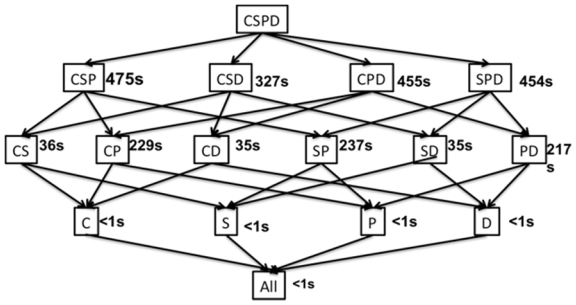


Figure 3: Computation time for each OLAP cuboid with M0 on single node (letters are dimension names: C=Customer, S=Supplier, D=Date, P=Part).

4.4 Document-oriented Data Warehouses versus Relational Data Warehouses

In this section, we compare loading times and querying between data warehouse instantiations on document-oriented and relational databases. In document-oriented systems, we consider the data model M0, because it performs better than the others. In the relational database, we consider two models R0 and R2 mentioned earlier. For R0, data is denormalized, everything is stored in one table: fact and dimension data. For R2, data is stored in a star-like schema i.e. the fact data is stored in one table and each dimension data is stored in a separate table.

Loading. First of all, we observe that relational databases demand for much less memory than document-oriented systems. Precisely, for scale factor sf=1, we need 15GB for data model M0 in MongoDB. Instead we need respectively 4.2GB and 1.2GB for data models R0 and R2 in PostgreSQL. This is easily explained. Document-oriented systems repeat field names on every document and specifically in MongoDB data types are also stored. To store data with flat models we need about 4 times more space, due to data redundancy. The same proportions are also observed on loading times.

Querying. We first compare query performance on the 4 query sets defined earlier (QS1, QS2, QS3, Q4) on a single node. We observe immmediately that queries run significantly faster on PostgreSQL (20 to 100 times). This is partly due to the relatively high

selectivity of the considered queries. Almost all data fits in memory.

Table 3: Query execution time per model, single node setting.

sf=1	M0	M1	M2	M3
Q1.1	*62s*	*62s*	*37s*	*94s*
Q1.2	*59s*	*61s*	*33s*	*91s*
Q1.3	*58s*	*58s*	*33s*	*86s*
Q1 avg	*60s*	*61s*	*34s* ✓	*90s*
Q2.1	*36s*	*39s*	*85s*	*105s*
Q2.2	*37s*	*41s*	*83s*	*109s*
Q2.3	*37s*	*40s*	*83s*	*109s*
Q2 avg	*37s* ✓	*40s*	*84s*	*108s*
Q3.1	*36s*	*36s*	*89s*	*100s*
Q3.2	*40s*	*40s*	*89s*	*104s*
Q3.3	*38s*	*38s*	*92s*	*104s*
Q3 avg	*38s* ✓	*38s*	*90s*	*103s*
Q4	*74s* ✓	*77s*	*689s*	*701s*

Table 4: Query execution time per model, cluster setting.

sf=1	M0	M1	M2	M3
Q1.1	*150s*	*152s*	*50s*	*129s*
Q1.2	*141s*	*142s*	*47s*	*125s*
Q1.3	*141s*	*141s*	*47s*	*127s*
Q1 avg	*144s*	*145s*	*48s* ✓	*127s*
Q2.1	*140s*	*140s*	*85s*	*107s*
Q2.2	*140s*	*142s*	*84s*	*103s*
Q2.3	*140s*	*138s*	*86s*	*111s*
Q2 avg	*140s*	*145s*	*85s* ✓	*107s*
Q3.1	*137s*	*138s*	*97s*	*105s*
Q3.2	*140s*	*143s*	*99s*	*107s*
Q3.3	*142s*	*143s*	*98s*	*108s*
Q3 avg	*139s*	*141s*	*98s*	*106s*
Q4	*173s* ✓	*180s*	*747s*	*637s*

In addition, we considered OLAP queries that correspond to the computation of OLAP cuboids. These queries are computationally more expensive than the queries considered previously (QS1, QS2, QS3, Q4). More precisely, we consider here the generation of OLAP cuboids on combinations of 3 dimensions. We call this query set QS5.

Average execution times on all query sets are shown in Table 5. We observe that the situation is reversed on this query set. Query execution times are comparable to each other. Queries run faster on MongoDB with data model R0 (singlenode) than on PostgresSQL. Queries run fastest on PostgreSQL with data model R2. MongoDB is faster if we consider the distributed setting.

Table 5: Average querying times by query set and approach.

single node sf=1	M0	R0	R2
QS1	*144s*	*7s*	*1s*
QS2	*140s*	*3s*	*2s*
QS3	*139s*	*3s*	*2s*
Q4	*173s*	*3s*	*1s*
QS5	*423s*	*549s*	*247s*

On these queries we have to keep in memory much more data than for queries in QS1, QS2, QS3 and QS4. Indeed, on the query sets QS1, QS2, QS3 and QS4 the amount of data to be processed is reduced by filters (equivalent of SQL where instructions). Then data is grouped on fewer dimensions (0 to 2). The result is fewer data to be kept in memory and fewer output records. Instead for computing 3 dimensional cuboids, we have to process all data and the output has more records. Data will not fit in main memory in MongoDB or PostgreSQL. Nonetheless MongoDB seems suffering less this aspect than PostgreSQL.

We can conclude that MongoDB scales better when the amount of data to be processed increases significantly. It can also take advantage of distribution. Instead, PostgresSQL performs very well when all data fits in main memory.

5 CONCLUSIONS

In this paper, we have studied the instantiation of data warehouses with document-oriented systems. For this purpose, we formalized and analyzed four logical models. Our study shows weaknesses and strengths across the models. We also compare the best performing data warehouse instantiation in document-oriented systems with 2 instantiations in relational database.

Depending on queries and data warehouse usage, we observe that the ideal model differs. Some models require less disk space, more precisely M2 and M3. This is due to the redundancy of data in models M0 and M1 that is avoided with models M2 and M3. For highly selective queries, we observe no ideal model. Queries run sometimes faster on one model and sometimes on another. The situation changes fast when queries are less selective. On data with models M2 and M3, we observe that querying suffers from joins. For queries that are poorly selective, we observe a significant impact on query execution times making these models non-recommendable.

We also compare instantiations of data warehouses on a document-oriented system with a relational system. Results show that RDBMS is faster on querying raw data. But performance slows down quickly when data does not fit on main memory. Instead, the analysed document-oriented system is shown more robust i.e. it does not have significant performance drop-off with scale increase. As well, it is shown to benefit from distribution. This is a clear advantage with respect to RDBMS that do not scale

well horizontally; they have a lower maximum database size than NoSQL systems.

In the near future, we are currently studying another document-oriented system and some column-oriented systems with the same objective.

ACKNOWLEDGEMENTS

This work is supported by the ANRT funding under CIFRE-Capgemini partnership.

REFERENCES

E. Annoni, F. Ravat, O. Teste, and G. Zurfluh. Towards Multidimensional Requirement Design. 8th International Conference on Data Warehousing and Knowledge Discovery (DaWaK 2006), LNCS 4081, p.75-84, Krakow, Poland, September 4-8, 2006.

A. Bosworth, J. Gray, A. Layman, and H. Pirahesh. Data cube: A relational aggregation operator generalizing group-by, cross-tab, and sub-totals. Tech. Rep. MSRTR-95-22, Microsoft Research, 1995.

M. Chevalier, M. El Malki, A. Kopliku, O. Teste, Ronan Tournier. Not Only SQL Implementation of multidimensional database. International Conference on Big Data Analytics and Knowledge Discovery (DaWaK 2015a), p. 379-390, 2015.

M. Chevalier, M. El Malki, A. Kopliku, O. Teste, R. Tournier. Implementation of multidimensional databases in column-oriented NoSQL systems. East-European Conference on Advances in Databases and Information Systems (ADBIS 2015b), p. 79-91, 2015.

M. Chevalier, M. El Malki, A. Kopliku, O. Teste, R. Tournier. Benchmark for OLAP on NoSQL Technologies. IEEE International Conference on Research Challenges in Information Science (RCIS 2015c), p. 480-485, 2015.

Chaudhuri and U. Dayal. An overview of data warehousing and OLAP technology. SIGMOD Record 26(1), ACM, pp. 65-74, 1997.

Colliat. OLAP, relational, and multidimensional database systems. SIGMOD Record 25(3), pp. 64.69, 1996.

Cuzzocrea, L. Bellatreche and I. Y. Song. Data warehousing and OLAP over big data: current

Dede, M. Govindaraju, D. Gunter, R.S. Canon and L. Ramakrishnan. Performance evaluation of a mongodb and hadoop platform for scientific data analysis. 4th ACM Workshop on Scientific Cloud Computing (Cloud), ACM, pp.13-20, 2013.

Dehdouh, O. Boussaid and F. Bentayeb. Columnar NoSQL star schema benchmark. Model and Data Engineering, LNCS 8748, Springer, pp. 281-288, 2014.

Floratou, N. Teletia, D. Dewitt, J. Patel and D. Zhang. Can the elephants handle the NoSQL onslaught? Int. Conf. on Very Large Data Bases (VLDB), pVLDB 5(12), VLDB Endowment, pp. 1712–1723, 2012.

Golfarelli, D. Maio and S. Rizzi. The dimensional fact model: A conceptual model for data warehouses. Int. Journal of Cooperative Information Systems 7(2-3), World Scientific, pp. 215-247, 1998.

S. Kanade and A. Gopal. A study of normalization and embedding in MongoDB. IEEE Int. Advance Computing Conf. (IACC), IEEE, pp. 416-421, 2014.

R. Kimball and M. Ross. The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling. John Wiley & Sons, 2013.

M. J. Mior. Automated schema design for NoSQL databases. SIGMOD PhD symposium, ACM, pp. 41-45, 2014.

P. ONeil, E. ONeil, X. Chen and S. Revilak. The Star Schema Benchmark and augmented fact table indexing. Performance Evaluation and Benchmarking, LNCS 5895, Springer, pp. 237-252, 2009.

F. Ravat, O. Teste, G. Zurfluh. A Multiversion-Based Multidimensional Model. 8th International Conference on Data Warehousing and Knowledge Discovery (DaWaK 2006), LNCS 4081, p.65-74, Krakow, Poland, September 4-8, 2006.

J. Schindler. I/O characteristics of NoSQL databases. Int. Conf. on Very Large Data Bases (VLDB), pVLDB 5(12), VLDB Endowment, pp. 2020-2021, 2012.

Zhao and X. Ye. A practice of TPC-DS multidimensional implementation on NoSQL database systems. Performance Characterization and Benchmarking, LNCS 8391, pp. 93-108, 2014.

Faceted Queries in Ontology-based Data Integration

Tadeusz Pankowski
Institute of Control and Information Engineering, Poznań University of Technology, Poznań, Poland
tadeusz.pankowski@put.poznan.pl

Keywords: Data Integration, Faceted Queries, Ontology, RDF Data.

Abstract: The aim of using ontology-based data integration is to provide users with a unified view, in a form of a global application domain ontology, over a multitude of data sources. The terminological component of this ontology is then presented as the global schema of the system and is used as the reference model for formulating queries. The extensional knowledge consists of RDF data sets (graphs) stored in local databases. In such scenario, a faceted query interface is a desired solution for end-user data access. Then there is a need for effective query answering utilizing both extensional and intentional knowledge representation. In this paper, we propose and discuss a possible solution to this issue. We show how a class of deductive rules, in particular Datalog rules and rules defining functionality, can be incorporated in the process of ontology-enhanced query answering in ontology-based data integration systems.

1 INTRODUCTION

Data integration systems provide users with a uniform view over a multitude of heterogeneous data sources. This uniform view has a form of a *global schema*, which realize so called global-as-view (GAV) paradigm (Lenzerini, 2002), (Ullman, 1997). The data is stored in data sources in their own schemas. The global schema frees users from having to locate the sources relevant to their queries. In order to answer queries formulated against the global schema, the system provides the semantic mappings between the global schema and the local (source) schemas (Halevy et al., 2006), (Calì et al., 2004), (Fagin et al., 2009), (Bernstein and Haas, 2008).

Currently, a broad class of data integration systems uses ontologies as global schemas, which led to emergence of *ontology-based data integration* (OBDI) and to *ontology-based data access* (OBDA) (Cruz and Xiao, 2009), (Calvanese et al., 2010), (Das et al., 2004), (Eklund et al., 2004), (Calvanese et al., 2007a), (Skjæveland et al., 2015).

Now, the most popular means to specify and query ontologies are OWL (OWL 2 Web Ontology Language Profiles, 2009), RDF (Resource Description Framework (RDF) Model and Syntax Specification, 1999) and SPARQL (SPARQL Query Language for RDF, 2008). OWL provides a method to formalize a domain by defining classes and properties of those classes (by means of *rules* or *axioms*), and to define individuals and assert properties about them (by means of *facts* or *assertions*). OWL is based on description logic (Baader et al., 2003). In practice, ontology rules are usually written in a form of first order language (FOL) formulas, and facts are usually defined by means of RDF graphs or FOL sentences. A standard language to formulate queries over ontologies is SPARQL. SPARQL, however, is not suitable query language for end-users. Instead, so called *faceted search* is used for end-user data access (Yee et al., 2003), (Oren et al., 2006), (Hahn et al., 2010).

In OBDI systems, an ontology is divided into two components: *terminological component* (TBox) consisting of *signature* (a set of unary predicate names (classes) and binary predicate names (properties)), and *rules* (axioms); and *assertional component* consisting of a set of *facts* (assertions). The terminological component forms the *global schema* of OBDI, and the assertional component consists of a set of local databases. Any local database state is a set of RDF data, which can be represented as an RDF graph, so we call it a *graph database*. The set of RDF graphs forms *extensional* knowledge. The set of rules in global schema constitutes the *intentional* knowledge about the application domain, and substantially enrich the extensional knowledge. The challenging issue in OBDI is to take into account both the extensional and intentional knowledge while answering queries. Data in different local databases can complement each other and can overlap. However, we as-

Pankowski, T.
Faceted Queries in Ontology-based Data Integration.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 150-157
ISBN: 978-989-758-187-8

sume that they are consistent with the global schema and do not contradict one another.

In this paper, we propose a method for query answering in OBDI systems in the situation when queries are formulated against the global schema as *faceted queries*. To create the answer, the service uses relevant data from all local databases as well as data necessary in reasoning procedures implied by ontology rules.

The main contribution of the paper, is the proposal of an algorithm for extending the query graph (created from a faceted query) with edges implied by relevant deductive rules. The set of considered ontology rules includes so called Datalog rules and rules specifying functionality of binary predicates and their inversions (key properties). We also show how the extended query graph (a *global query pattern*) can be used to merge local answers by means of a chase procedure.

The structure of the paper is as follows. Some preliminaries concerning graph databases, ontologies and queries, are reviewed in Section 2. Faceted queries are defined in Section 3, and their formal semantics, understood as first order open formulas, is given. In Section 4 we characterize ontology-based data integration, and describe architecture of an OBDI system. Also the running example is introduced. The process of answering faceted queries is described in Section 5. We propose an algorithm for creating global, ontology-enhanced query graph, and its usage in merging local answers and obtaining the final result. Section 6 concludes the paper.

2 PRELIMINARIES

A *graph database* is a finite, edge-labeled and directed graph (Barceló and Fontaine, 2015). Formally, let the following sets be infinite and pairwise disjoint: Const – a set of *constants*, $\mathsf{LabNull}$ – a set of *labeled nulls* (treated as variables), UP – a set of *unary predicates*, BP – set of *binary predicates*. Additionally, we assume that type and $\approx$ are distinguished binary predicates in BP. A *signature* Σ is a finite subset of $\mathsf{UP} \cup \mathsf{BP}$.

A *graph database* (or *RDF graph*) $G = (V, E)$ with signature Σ consists of a finite set $V \subseteq \mathsf{Const} \cup \mathsf{LabNull} \cup \mathsf{UP}$ of *node identifiers* (or *nodes* for short) and a finite set of labeled edges (or *facts*) $E \subseteq V \times \Sigma \times V$, such that:

- if $(v_1, p, v_2) \in E$ and $p \in \mathsf{BP} \setminus \{\mathsf{type}\}$, then $v_1, v_2 \in \mathsf{Const} \cup \mathsf{LabNull}$,
- if $(v_1, \mathsf{type}, v_2) \in E$ then $v_1 \in \mathsf{Const} \cup \mathsf{LabNull}$, and $v_2 \in \mathsf{UP}$.

In first order logic (FOL), we use the following notation for edges:

- for (v, type, C), where $C \in \mathsf{UP}$, we use $C(v)$,
- for $(v_1, \approx, v_2)$ we use $v_1 \approx v_2$,
- for $(v_1, P, v_2$, where $P \in \mathsf{BP}$, we use $P(v_1, v_2)$.

A *rule* is a FOL sentence (implication) of the form $\forall \mathbf{x} \forall \mathbf{y}(\varphi(\mathbf{x}, \mathbf{y}) \rightarrow \exists \mathbf{z} \psi(\mathbf{x}, \mathbf{z}))$, where $\mathbf{x}, \mathbf{y}, \mathbf{z}$ are tuples of variables. Formulas φ (the *body*) and ψ (the *head*) are conjunctions of atoms of the form $C(v)$, $P(v_1, v_2)$, and $v_1 \approx v_2$, where v, v_1, v_2 ranges over $\mathsf{Const} \cup \mathsf{LabNull}$. If the tuple $\mathbf{z}$ of existentially quantified variables is empty, the rule is *Datalog rule*. By $G \cup R$ we denote all facts belonging to G and deduced from G using rules in R.

An *ontology* (or a *knowledge base*) is a triple $O = (\Sigma, R, G)$, where Σ is a signature, R is a finite set of rules, and G is a database graph (a set of facts). A kind of ontology depends on the form of rules. For example, OWL 2 defines three profiles with different computational properties (OWL 2 Web Ontology Language Profiles, 2009).

A *query* is a FOL open formula. If the formula is constructed only with: (a) atoms of the form $C(v)$, $P(v_1, v_2)$ and $v \approx a$, where v, v_1, v_2 are variables, and a is a constant or labeled null; (b) symbols of conjunction ($\wedge$), disjunction ($\vee$), and existential quantification ($\exists$), then the query is a *positive existential query* (PEQ). A PEQ is *monadic* if has exactly one free variable, and is *conjunctive query* (CQ) if disjunction does not occur in this query.

A query $Q(\mathbf{x})$, where $\mathbf{x}$ is a tuple of free variables, is *satisfiable* in $O = (\Sigma, R, G)$, denoted $O \models Q(\mathbf{x})$ if Q is built from predicates in Σ, and there is a tuple $\mathbf{a}$ of elements from $\mathsf{Const} \cup \mathsf{LabNull}$ such that $G \cup R \models Q(\mathbf{a})$. Then $\mathbf{a}$ is an *answer* to $Q(\mathbf{x})$ with respect to O. Set of answers will be denoted by $Ans(Q)$.

3 FACETED QUERIES

There is an increasing number of data centered systems based on RDF and OWL 2. A standard query languages in such systems is SPARQL. This language, however, is not a convenient to end-users. As a more suitable interface for end-user data access have been developed approaches based on so-called *faceted search*. Now, we will define *faceted queries* for search over RDF graphs, and we will consider answering such queries when a database is additionally enhanced with an ontology. The considered system is a data integration system, where the graph data must be composed from data graphs stored in local databases.

In (Arenas et al., 2014), a *facet* is defined as a pair: $F = (X, \wedge\Gamma)$ (*conjunctive* facet), or $F = (X, \vee\Gamma)$ (*disjunctive* facet), where:

- $X \in \mathsf{BP}$ is the *facet name*, denoted by $F|_1$,
- Γ defines a set of *facet values* and is denoted $F|_2$,
- if $X = \mathsf{type}$, then $\Gamma \subseteq \mathsf{UP}$,
- if $X \in \mathsf{BP} \setminus \{\mathsf{type}\}$, then $\Gamma \subseteq \mathsf{Const} \cup \{\mathsf{any}\}$ or $\Gamma \subseteq \mathsf{UP} \cup \{\mathsf{any}\}$.

Any faceted query can be represented by a user-friendly graphical interface. A graphical form of faceted query in Figure 1, searches for ACM authors from NY university who have written a publication in year 2014.

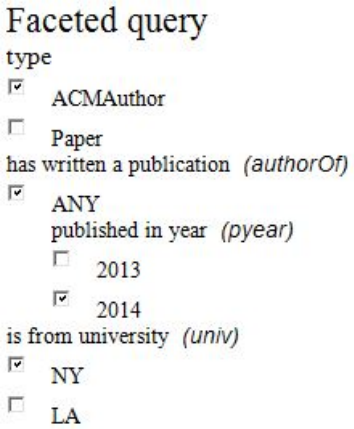


Figure 1: A graphical form of a faceted query.

Example 3.1. *For the considered example, the following facets can be defined:*

$$F_1 = (\mathsf{type}, \vee\{ACMAuthor, Paper\}),$$
$$F_2 = (authorOf, \vee\{\mathsf{any}, p_1, a_1, a_2\}),$$
$$F_3 = (pyear, \vee\{\mathsf{any}, 2013, 2014\}),$$
$$F_4 = (univ, \vee\{\mathsf{any}, NY, LA\}),$$
$$F_5 = (uinv, \wedge\{NY, LA\}).$$

Note, that F_5 is a conjunctive facet and denotes individuals which are simultaneously from two universities – NY and LA.

Definition 3.2. *Let $F = (X, \circ\Gamma)$, $\circ \in \{\wedge, \vee\}$, be a facet. A basic faceted query determined by F is a pair of the form $Q = (X, S)$, where $S \subseteq \Gamma$. A basic faceted query will be denoted by Q_t, if $X = \mathsf{type}$, and by Q_b when $X \in \mathsf{BP} \setminus \{\mathsf{type}\}$. A faceted query (or query for short) is an expression Q conforming to the following grammar:*

$$Q ::= q \mid (q \wedge q) \mid (q \vee q)$$
$$q ::= Q_t \mid Q_b \mid (Q_b / Q)$$

Example 3.3. *The faceted query corresponding to this in Figure 1 is:*

$$Q = ((F_1, \{ACMAuthor\}) \wedge (F_4, \{NY\})) \\ \wedge((F_2, \{\mathsf{any}\}) / (F_3, \{2014\})) \quad (1)$$

In Definition 3.4, we define semantics for faceted queries. The semantics $[\![Q(x)]\!]$ assigns to each query Q and a given variable x, a monadic PEQ with one free variable x.

Definition 3.4. *Let Q_t be a basic faceted query over $F_t = (\mathsf{type}, \circ\Gamma)$, Q_b be a basic faceted query over $F_P = (P, \circ\Gamma)$, $P \in \mathsf{BP} \setminus \{\mathsf{type}\}$, and Q be an arbitrary faceted query. Then semantics of faceted queries is defined as follows:*

1. $Q_t = (F_t, S)$, $S \subseteq \mathsf{UP}$*:*
$$[\![Q_t(x)]\!] = \underset{C \in S}{\circ} C(x).$$
2. $Q_b = (F_P, \{\mathsf{any}\})$*:*
$$[\![Q_b(x)]\!] = \exists y\, P(x,y),$$
$$[\![(Q_b/Q)(x)]\!] = \exists y\, P(x,y) \wedge [\![Q(y)]\!].$$
3. $Q_b = (F_P, S)$, $S \subseteq \mathsf{Const} \cup \mathsf{LabNull}$*:*
$$[\![Q_b(x)]\!] = \underset{a_i \in S}{\circ} \exists y_i\, P(x,y_i) \wedge y_i \approx a_i,$$
$$[\![(Q_b/Q)(x)]\!] = \underset{a_i \in S}{\circ} \exists y_i\, P(x,y_i) \wedge y_i \approx a_i \wedge [\![Q(y_i)]\!].$$
4. $Q_b = (F_P, S)$, $S \subseteq \mathsf{UP}$*:*
$$[\![Q_b(x)]\!] = \underset{C_i \in S}{\circ} \exists y_i\, P(x,y_i) \wedge C_i(y_i),$$
$$[\![(Q_b/Q)(x)]\!] = \underset{C_i \in S}{\circ} \exists y_i\, P(x,y_i) \wedge C_i(y_i) \wedge [\![Q(y_i)]\!].$$
5. $Q_b = (F_P, \{\mathsf{any}\} \cup S)$*, :*
$$[\![Q_b(x)]\!] = [\![(F_P, \{\mathsf{any}\})(x)]\!] \circ [\![(F_P, S)(x)]\!],$$
$$[\![(Q_b/Q)(x)]\!] = [\![((F_P, \{\mathsf{any}\})/Q)(x)]\!] \\ \circ [\![((F_P, S)/Q)(x)]\!].$$
6. *Let q_1 and q_2 be queries, then:*
$$[\![(q_1 \wedge q_2)(x)]\!] = [\![q_1(x)]\!] \wedge [\![q_2(x)]\!],$$
$$[\![(q_1 \vee q_2)(x)]\!] = [\![q_1(x)]\!] \vee [\![q_2(x)]\!].$$

□

The semantics of basic type-faceted queries of the form (F, S) is the conjunction (disjunction) of atoms of the form $C(x)$ over the same variable, where $C \in S$. If the facet name is a binary predicate P, then the query is translated to: (a) an atom whose second argument is existentially quantified (if any occurs); (b) a conjunction (disjunction) of binary atoms whose second argument must be equal to a constant or a labeled null, or must satisfy an unary predicate. In the case of nesting, a variable from the parent is shifted to the

child. Finally, conjunction (disjunction) of queries is interpreted as the conjunction (disjunction) of the corresponding formulas.

The first order interpretation (the PEQ) of faceted query (1) is given in (2).

$$\begin{aligned}[\![Q(x)]\!] = \quad & ACMAuthor(x) \\ & \wedge \exists y(authorOf(x,y) \\ & \quad \wedge \exists z(pyear(y,z) \wedge z \approx 2014)) \\ & \wedge \exists w(univ(x,w) \wedge w \approx NY).\end{aligned} \tag{2}$$

4 ONTOLOGY-BASED DATA INTEGRATION

Ontology Based Data Integration (OBDI), or Ontology Based Data Access (OBDA) involves the use of ontology to effectively combine data or information from multiple heterogeneous sources (Wache et al., 2001). In this paper we will follow so called *single ontology approach*, i.e., an approach when a single ontology is used as a global reference model in the system. We assume that the data integration system is based on a global schema (Ullman, 1997), (Halevy et al., 2006), (Lenzerini, 2002) (another approaches assume P2P data integration, see for example (Calvanese et al., 2004)).

On the conceptual level, a user perceives contents of the system as a large single ontology $O = (\Sigma, R, G)$, and formulates faceted queries against this ontology. On the implementation level, we assume that (see Figure 2):

1. The (global) schema of the system consists of the signature and the set of rules of the ontology, i.e., $Sch = (\Sigma, R)$.
2. Facts, represented by means of RDF graphs, are stored in local databases, $DB_i = (\Sigma_i, G_i)$, where $\Sigma_i \subseteq \Sigma$ consists of symbols, i.e., unary and binary predicates, occurring in RDF graph G_i.
3. Data in different local databases complement each other and can overlap. We assume, however, that databases are consistent and do not contradict one another.
4. Local databases are created by local users and the global schema is used as the reference in introducing new facts.

A user query is rewritten and sent to each database in a form understandable and executable to this database management system. Next, partial answers are sent back, merged accordingly and finally returned to the user.Answering queries requires some data inferring processes implied by ontology deductive rules. The architecture of such a system is given in Figure 2.

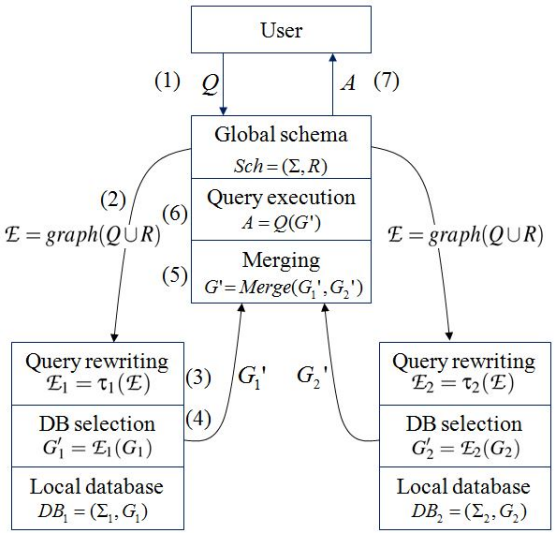


Figure 2: Architecture of an ontology-based data integration.

The system works as follows:

(1) The user formulates a faceted query Q.

(2) Q is translated to a FOL formula $[\![(Qx)]\!]$ and its graph representation is created. This graph is extended to $\mathcal{E}$ with elements corresponding to relevant deductive rules in the schema. Graph $\mathcal{E}$ is sent to local database management systems.

(3) $\mathcal{E}$ is reduced to $\mathcal{E}_i$ using information from signature Σ_i.

(4) A set of edges is selected from G_i, which are relevant to query answering.

(5) Selected subgraphs are merged into graph G'.

(6) The user query Q is evaluated over G', and the answer A is obtained.

(7) A is returned to the user.

Example 4.1. *The global schema, $O = (\Sigma, R)$, relevant to our example can contain the following set of deductive rules:*

$$\begin{aligned}
&(R1) \quad atConf(x,y) \wedge y \approx ACMConf \\
&\qquad \rightarrow ACMPaper(x), \\
&(R2) \quad authorOf(x,y) \wedge ACMPaper(y) \\
&\qquad \rightarrow ACMAuthor(x), \\
&(R3) \quad atConf(x,y) \wedge cyear(y,z) \rightarrow pyear(x,z), \\
&(R4) \quad univ(x,y_1) \wedge univ(x,y_2) \rightarrow y_1 \approx y_2, \\
&(R5) \quad title(x_1,y) \wedge title(x_2,y) \rightarrow x_1 \approx x_2, \\
&(R6) \quad Author(x) \rightarrow \exists y(authorOf(x,y) \wedge Paper(y)).
\end{aligned}$$

Sample RDF graphs, G_1 and G_2, of two local databases are given in, respectively, Figure 3 and Figure 4. These graphs are built over signatures, respectively, Σ_1 and Σ_2, being subsets of Σ, and over a set of constants, Const, and a set of labeled nulls LabNull.

In this case $John, Ann, 2014, 2013, KB, AI, NY, LA$, $ACMConf$, and $IEEEConf$ are in Const, and p_1, a_1, a_2 are in LabNull. Labeled nulls are used as identifiers of anonymous nodes and can be replaced with other labeled nulls or with constants. So, they are like variables (Fagin et al., 2005).

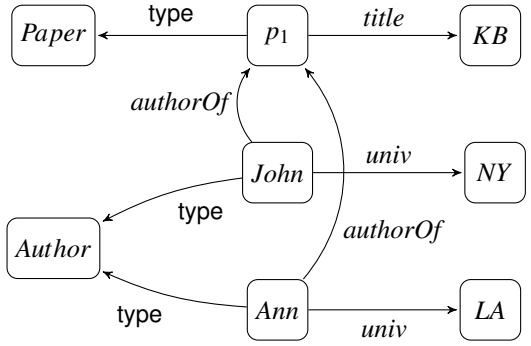


Figure 3: RDF graph G_1 of a local database DB_1.

The FOL form of G_1:

$$Paper(p_1), title(p_1, KB), Author(John), Author(Ann), authorOf(John, p_1), authorOf(Ann, p_1), univ(John, NY), univ(Ann, LA),$$

and of G_2:

$$Paper(a_1), Paper(a_2), title(a_1, KB), atConf(a_1, ACMConf), cyear(ACMConf, 2014), title(a_2, AI), atConf(a_2, IEEEConf), cyear(IEEEConf, 2013), Author(Ann), authorOf(Ann, a_1), authorOf(Ann, a_2).$$

If we evaluate query (2) against G_1 or/and G_2, then the answer is empty (in particular, the binary relation *ACMAuthor* does not even exist). However, if we consider also the set of rules in the ontology and apply them to infer new facts, we see that (2) is satisfied by *John*. So *John* is the answer to the query under consideration. Thus, to obtain the answer we have to:

- merge database states,
- take into account deductive rules from the ontology,
- apply deductive rules to infer new facts from the result of merging,
- evaluate the query over the set of all facts.

Note, however, that a naive performance of these operations can be rather inefficient. For example, we can merge whole database states – which is rather very inefficient, or we can take into consideration only such subgraphs which are relevant to obtain the answer. Further on in the paper, we will discuss how these relevant subgraphs can be chosen.

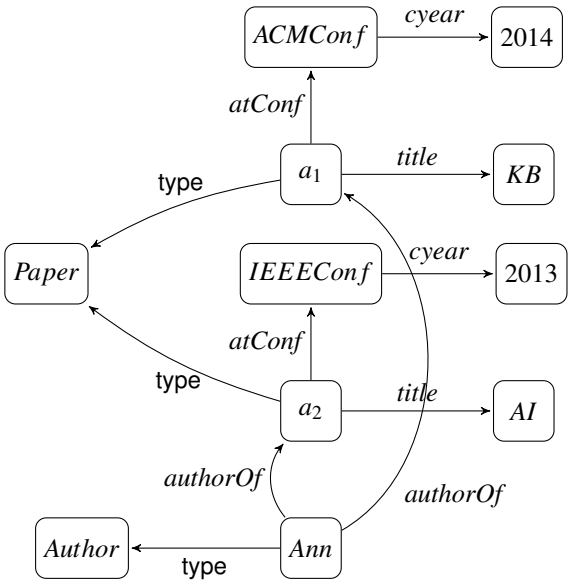


Figure 4: RDF graph G_2 of a local database DB_2.

5 ANSWERING FACETED QUERIES

5.1 Creating Global Query Patterns

Now, we will discuss the problem of selecting some facts (edges) from RDF graphs (step (4) in Figure 2) which should be sent to the merge stage (step (5) in Figure 2). The general assumption about the selection is that there must be *justification* to select an edge. The selection of an edge (x, P, y) is justified if:

- predicate P occurs in the query;
- predicate P occurs in the left hand site of an ontology rule, and there is a justification to select a predicate P' occurring on the right hand site of this rule;
- P is functional or a key (P^- is functional) and can be used to infer an equality between some data involved in the answer to the query.

A facet graph $\mathcal{G}$ for a facet query Q represented by a FOL formula $[\![Q(x)]\!]$, is the graph $\mathcal{G} = (V, E)$, where:

1. V is a set of unary predicate names, variable names, constants and labeled nulls, occurring in

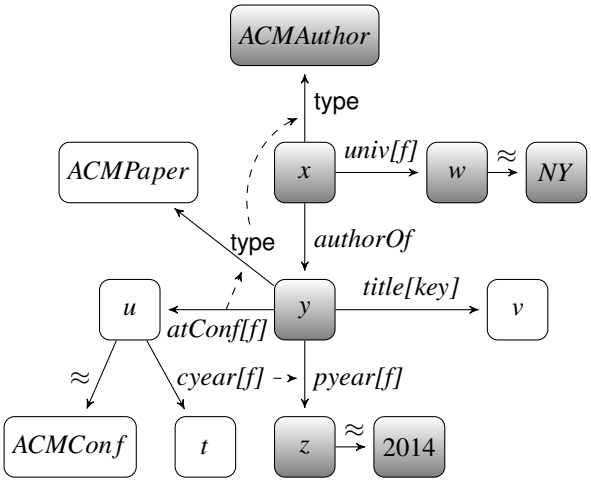


Figure 5: Extended query graph $\mathcal{E}$.

$[\![Q(x)]\!]$.

2. The set E of edges is defined as follows:
   - if $C(v)$ is in $[\![Q(x)]\!]$, then (v, type, C) is in E,
   - if $P(v_1, v_2)$ is in $[\![Q(x)]\!]$, then (v_1, P, v_2) is in E,
   - if $v \approx a$ is in $[\![Q(x)]\!]$, then $(v, \approx, a)$ is in E.

In Figure 5, the subgraph with greyed nodes constitutes the query graph for the faceted query in Figure 1 and its FOL interpretation (2). Additionally, in Figure 5 some edges are qualified with: $[f]$, to denote that the corresponding binary predicate is a *function* (rule $(R4)$), and $[key]$, to denote that the corresponding binary predicate is a *key*, i.e., its inversion is a function (rule $(R5)$).

Next, the query graph $\mathcal{G}$ is extended to an *extended query graph* (or *global query pattern*), $\mathcal{E} = (V_\mathcal{E}, E_\mathcal{E})$, by adding some edges implied by ontology rules. We take into account rules which are adjacent to the current form of the extended query graph. We proceed as follows:

1. We start with assuming $\mathcal{E} = (V_\mathcal{E}, E_\mathcal{E})$ equal to $\mathcal{G} = (V, E)$.
2. Let $\varphi \to C(v)$ be a rule and (x, type, C) be in $E_\mathcal{E}$, for some variable x. Then:
   - rename all variables occurring in φ, with the exception of variable v, in such a way that new names are different from those occurring in $V_\mathcal{E}$;
   - rename v to x,
   - match the renamed form of φ to edges in $E_\mathcal{E}$, and rename variables accordingly. The result denote by φ',
   - extend $E_\mathcal{E}$ as follows:
     - if $C(w)$ is in φ' and not in $E_\mathcal{E}$, then add the edge (w, type, C) to $E_\mathcal{E}$,
     - if $P(w_1, w_2)$ is in φ' and not in $E_\mathcal{E}$, then add (w_1, P, w_2) to $E_\mathcal{E}$,
     - if $w \approx a$ is in φ' and not in $E_\mathcal{E}$, add $(w, \approx, a)$ to $E_\mathcal{E}$.
3. Let $\varphi \to P(v_1, v_2)$ be a rule and (x, P, y) be in $E_\mathcal{E}$, for some variables x and y. Then:
   - rename all variables occurring in φ, with the exception of variables v_1 and v_2, in such a way that new names are different from those occurring in $V_\mathcal{E}$;
   - rename v_1 to x, and v_2 to y,
   - match the renamed form of φ to edges in $E_\mathcal{E}$, and rename variables accordingly. The result denote by φ',
   - extend $E_\mathcal{E}$ analogously to the extension described in (2).
4. Let φ be a rule defining functionality of a binary predicate $P(v_1, v_2)$. Let x be a variable in $V_\mathcal{E}$ defined over the range of P. Then rename v_2 to x, and v_1 to an appropriate name w, and add (w, P, x) to $E_\mathcal{E}$.
5. Let φ be a rule defining functionality of inversion of a binary predicate $P(v_1, v_2)$, i.e., determining that P is a key. Let x be a variable in $V_\mathcal{E}$ defined over the domain of P. Then rename v_1 to x, and v_2 to an appropriate name w, and add (x, P, w) to $E_\mathcal{E}$.

In Figure 5, edges with white nodes were added according to the above procedure. Dashed arrows indicate which edges are needed to infer another edges. In particular, $(y, \mathsf{type}, \mathit{ACMPaper})$ is necessary to infer $(x, \mathsf{type}, \mathit{ACMAuthor})$ (rule $(R2)$). To infer $(y, \mathsf{type}, \mathit{ACMPaper})$, we need (y, atConf, u) and $(u, \approx, \mathit{ACMConf})$ (rule $(R1)$). To infer (y, pyear, z), the edge (u, cyear, t) is needed, (rule $(R3)$). Finally, (y, title, v) is added since title is a *key*, i.e., its inversion, title^-, is a function (rule $(R5)$).

5.2 Local Answers to Graph Patterns

Restrictions of graph $\mathcal{E}$ (Figure 5) to DB_1 and DB_2 are extended graphs (*local query patterns*), $\mathcal{E}_1$ and $\mathcal{E}_2$, presented in Figure 6 and Figure 7, respectively.

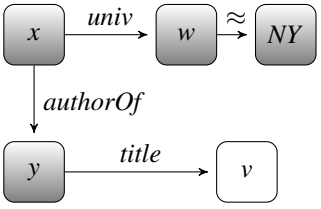


Figure 6: Extended query graph $\mathcal{E}_1 = \tau_{\Sigma_1}(\mathcal{E})$.

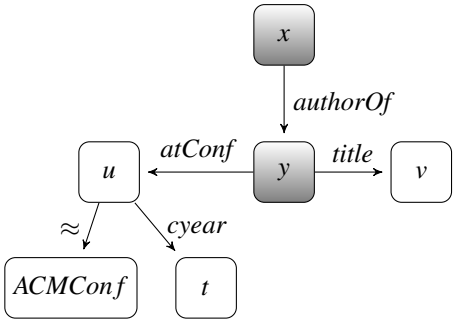


Figure 7: Extended query graph $\mathcal{E}_2 = \tau_{\Sigma_2}(\mathcal{E})$.

Subgraphs $G'_1 = \mathcal{E}_1(G_1)$ and $G'_2 = \mathcal{E}_2(G_2)$, which are answers to pattern queries $\mathcal{E}_1$ and $\mathcal{E}_2$, respectively, are presented in Figure 8 and Figure 9, respectively.

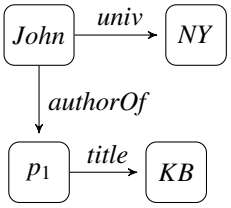


Figure 8: Answer $G'_1 = \mathcal{E}_1(G_1)$.

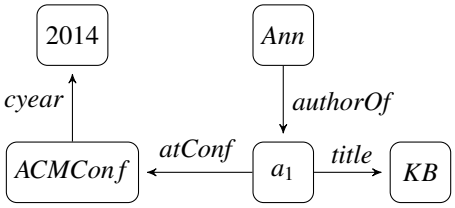


Figure 9: Answer $G'_2 = \mathcal{E}_2(G_2)$.

Next, RDF subgraphs G'_1 and G'_2, are sent to the merging service.

5.3 Merging Local Answers

Partial answers, like G'_1 and G'_2, must be merged to produce a RDF graph over which the user query Q can be evaluated. Now, we propose a method to perform the merging. The merge is done by means of mapping rules produced from the extended query graph $\mathcal{E}$ and from the set R of ontology rules belonging to the global schema. These rules are used to define the *chase procedure* as it was proposed in data exchange theory (Fagin et al., 2005), (Calvanese et al., 2007b). Predicates prefixed by s refer to source data, i.e., to G'_1 and G'_2. Predicates without prefixes, refer to target data, i.e., to the result of the merge, and are understood as targed constraints. In our case, the set of generated mapping rules used for merging is given in Figure 10.

$$
\begin{aligned}
&s.authorOf(x,y) \rightarrow authorOf(x,y),\\
&s.univ(x,y) \rightarrow univ(x,y),\\
&s.title(x,y) \rightarrow title(x,y),\\
&s.atConf(x,y) \wedge y \approx ACMConf \rightarrow ACMPaper(x),\\
&authorOf(x,y) \wedge ACMPaper(y) \rightarrow ACMAuthor(x),\\
&s.atConf(x,y) \wedge s.cyear(y,z) \rightarrow pyear(x,z),\\
&title(x_1,y) \wedge title(x_2,y) \rightarrow x_1 \approx x_2.
\end{aligned}
$$

Figure 10: Mapping rules used in merging.

In particular, the last rule enforces $a_1 \approx p_1$. So, in the result RDF graph all occurrences of a_1 are replaced by p_1. The result of merge is given in Figure 11.

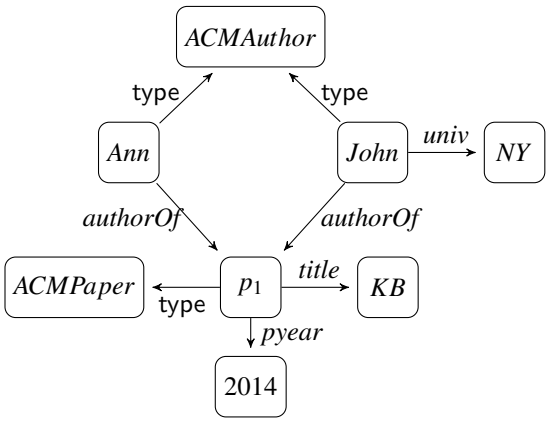


Figure 11: Result of merging, $G' = Merge(G'_1, G'_2)$, by means of mapping rules from Figure 10.

5.4 Obtaining Final Answers

The result of merging of local answers to local graph patterns, as $G' = Merge(G'_1, G'_2)$ in Figure 11, constitutes a dataset which is the object to evaluate a faceted query under consideration. The first order representation of the query, in our case (2), is a monadic PEQ resulting from a faceted query. So, the answer can be found in polynomial time. It is easily seen that the answer is *John*.

So called *refocussing* functionality in faceted queries allows for changing the free variable of the query Q. In consequence, the answer consists of all valuations of this free variable. In our example, if we want to now information about papers written by ACM authors, we should refocus our attention to the variable being the second argument of $authorOf$ predicate.

6 CONCLUSION

In this paper, we have discussed an ontology-based

data integration system with faceted query interface. In such a system we have both, *extensional* and *intentional* knowledge. The extensional knowledge is stored as RDF graphs in local databases, and the intentional knowledge is given as a set of rules constituting a set of axioms of a global ontology. A user formulates faceted queries in a user-friendly way using a simple graphical interface. Next, local databases are queried about data which is indirectly or directly (to infer new facts by means of ontology rules) necessary to answer the query. The set of local answers are merged and finally the expected answer is obtained. The proposed method is a base to introduce new functionality into our system of data integration.

REFERENCES

Arenas, M., Grau, B. C., Kharlamov, E., Marciuska, S., and Zheleznyakov, D. (2014). Faceted search over ontology-enhanced RDF data. In *ACM CIKM 2014*, pages 939–948. ACM.

Baader, F., Calvanese, D., McGuinness, D., Nardi, D., and Petel-Schneider, P., editors (2003). *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press.

Barceló, P. and Fontaine, G. (2015). On the data complexity of consistent query answering over graph databases. In *ICDT 2015*, volume 31 of *LIPIcs*, pages 380–397. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik.

Bernstein, P. A. and Haas, L. M. (2008). Information integration in the enterprise. *Commun. ACM*, 51(9):72–79.

Calì, A., Calvanese, D., Giacomo, G. D., and Lenzerini, M. (2004). Data integration under integrity constraints. *Information Systystems*, 29(2):147–163.

Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Poggi, A., and Rosati, R. (2007a). Ontology-based database access. In *SEBD 2007*, pages 324–331.

Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R., and Ruzzi, M. (2010). Using OWL in Data Integration. In *Semantic Web Information Management. Chapter 17*, pages 397–424. Springer.

Calvanese, D., Giacomo, G. D., et al., (2007b). EQL-Lite: Effective First-Order Query Processing in Description Logics. In *IJCAI, International Joint Conference on Artificial Intelligence*, pages 274–279.

Calvanese, D., Giacomo, G. D., Lenzerini, M., and Rosati, R. (2004). Logical Foundations of Peer-To-Peer Data Integration. In *PODS*, pages 241–251.

Cruz, I. F. and Xiao, H. (2009). Ontology driven data integration in heterogeneous networks. In *Complex Systems in Knowledge-based Environments*, pages 75–98.

Das, S., Chong, E., Eadon, G., and Srinivasan, J. (2004). Supporting Ontology-Based Semantic Matching in RDBMS. In *Proc. of the 30th International Conference on Very Large Data Bases, VLDB 2004, Toronto, Canada*, pages 1054–1065.

Eklund, P. W., II, R. J. C., and Roberts, N. (2004). Retrieving and exploring ontology-based information. In Staab, S. and Studer, R., editors, *Handbook on Ontologies*, pages 405–414. Springer.

Fagin, R., Haas, L. M., Hernández, M. A., Miller, R. J., Popa, L., and Velegrakis, Y. (2009). Clio: Schema mapping creation and data exchange. In *Conceptual Modeling: Foundations and Applications*, volume LNCS **5600**, pages 198–236.

Fagin, R., Kolaitis, P. G., Miller, R. J., and Popa, L. (2005). Data exchange: semantics and query answering. *Theor. Comput. Sci*, 336(1):89–124.

Hahn, R., Bizer, C., Sahnwaldt, C., Herta, C., Robinson, S., Bürgle, M., Düwiger, H., and Scheel, U. (2010). Faceted Wikipedia Search. In *BIS 2010*, volume 47 of *Lecture Notes in Business Information Processing*, pages 1–11. Springer.

Halevy, A. Y., Rajaraman, A., and Ordille, J. J. (2006). Data integration: The teenage years. In Dayal, U., Whang, K.-Y., Lomet, D. B., Alonso, G., Lohman, G. M., Kersten, M. L., Cha, S. K., and Kim, Y.-K., editors, *VLDB*, pages 9–16. ACM.

Lenzerini, M. (2002). Data integration: A theoretical perspective. In Popa, L., editor, *PODS*, pages 233–246. ACM.

Oren, E., Delbru, R., and Decker, S. (2006). Extending faceted navigation for RDF data. In *ISWC 2006*, volume 4273 of *Lecture Notes in Computer Science*, pages 559–572. Springer.

OWL 2 Web Ontology Language Profiles (2009). www.w3.org/TR/owl2-profiles.

Resource Description Framework (RDF) Model and Syntax Specification (1999). www.w3.org/TR/PR-rdf-syntax/.

Skjæveland, M. G., Giese, M., Hovland, D., Lian, E. H., and Waaler, A. (2015). Engineering ontology-based access to real-world data sources. *J. Web Sem.*, 33:112–140.

SPARQL Query Language for RDF (2008). http://www.w3.org/TR/rdf-sparql-query.

Ullman, J. D. (1997). Information integration using logical views. in: Database Theory - ICDT 1997. *Lecture Notes in Computer Science*, 1186:19–40.

Wache, H., Vgele, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H., and Hbner, S. (2001). Ontology-Based Integration of Information - A Survey of Existing Approaches. In *IJCAI 2001*, pages 108–117.

Yee, K.-P., Swearingen, K., Li, K., and Hearst, M. (2003). Faceted metadata for image search and browsing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '03, pages 401–408. ACM.

Towards an SDLC for Projects Involving Distributed Systems

Rodrigo Augusto dos Santos, Avelino F. Zorzo and Sabrina Marczak
Faculdade de Informática, Pontifícia Universidade Católica do Rio Grande do Sul, Ipiranga Avenue, Porto Alegre, Brazil
rodrigo.augusto@acad.pucrs.br, {avelino.zorzo, sabrina.marczak}@pucrs.br

Keywords: Distributed Systems, Distributed Teams, Project Management, Life Cycle, SDLC, PLC.

Abstract: Since the 1970's, Distributed Systems have been turning into a more viable and reliable option for the implementation of information systems. This evolution continued ever since, and now they are applicable to a variety of purposes, such as online games, cloud computational solutions, etc. It is possible then to assume that today, Distributed Systems are found everywhere, and that there is a great probability for any given in-progress software development project to be using this paradigm as part of its delivery. Thus, it is relevant to study the impacts that Distributed Systems bring to Project Management. In this paper we discuss those impacts and challenges, as well as propose a Software Development Lifecycle and some associated practices that are to be used for software development projects involving Distributed Systems. Such practices are optimized for implementation under a Waterfall model, but are also adaptable for use with well known agile framework Scrum. The preliminary validation with industry professionals suggests that our proposals do support more appropriate management and execution of projects involving Distributed Systems solutions.

1 INTRODUCTION

A project is defined by PMI (2015) as being "a temporary endeavour in that it has a defined beginning and end in time, and therefore defined scope and resources". According to PMBoK (2013), "Project Management is the application of knowledge, skills, tools and techniques to project activities to meet the project requirements".

By 1970, the wide adoption of Distributed Systems (DS) became a fact, and Information Technology (IT) Project Managers around the world were forced to deal with it. DS, according to Couloris et al (2012), "are the ones in which hardware or software components, located at networked computers, communicate and coordinate their actions only by passing messages".

Couloris et al (2012) also provides some examples that fit this definition, such as web search, multiplayer online games, and financial trading systems, thus stating that DS includes "many of the most significant technological developments of recent years", "ranging from a small intranet to the Internet". This obviously turns the intersection between PM and DS into a relevant research area.

Our hypothesis though is that system distribution in a project may be regularly "abstracted" by IT project teams, with decisions regarding it becoming delegated to development teams only. The rest of the project team would focus on supposedly "attention-worthy, value-driven requirements", such as screens, reports, and other "tangible" features, thus, greatly increasing the risk of project failure.

This abstraction culture would also reflect upon academia, with small attention from researches on the intersection of DS and PM. In order to shed some light into our hypothesis, we performed a Systematic Mapping Study (SMS) (Section 2.1), seeking to understand how the intersection between DS and PM has been studied in academia. We also performed an interview-based field study (Section 2.2) to understand industry's perception about the topic.

The results from the SMS and field-based study led us to propose a Software Devolopment Life Cycle (SDLC) and some practices associated with it, both tailored for Software Develpment Projects involving DS (Section 3). These proposals were preliminarily validated through the process of member checking (Section 3.2). The limitations and future work are described in Section 3.2 and Section 4, respectively.

2 RESEARCH BACKGROUND

In this section we present the methodologies used in our Systematic Mapping Study (SMS) and interview-based field study, as well as their results.

Santos, R., Zorzo, A. and Marczak, S.
Towards an SDLC for Projects Involving Distributed Systems.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 158-165
ISBN: 978-989-758-187-8

2.1 Systematic Mapping Study

As a comparison measure for the volume of research on DS PM, we have used PM involving Distributed Teams (DT). Since DT has been adopted by many organizations distributing their software development projects worldwide, seeking cost and quality advantages (Herbsleb,2001), DT PM became a popular research topic.

Although DS and DT are two distinct subjects, with no direct relation between them, both topics are present in a great number of today's IT projects, having the research on each of them the same characteristic of being able to intersect with PM. The SMS, thus, was performed for confirming the level of attention provided to DS PM when compared to the volume of studies focusing on DT PM.

The number of papers selected as a result of systematic search was 37 out of 127. Out of these, 28 focused on PM intersection with DT, 8 focused on PM intersection with DS and only 1 focused on PM intersecting both DT and DS at the same time. These results demonstrate an imbalance in the academic interest towards both DT and DS. Another imbalance indicator is that out of the 8 DS PM papers, 50% of them were published before year 2000.

2.2 Interview-based Field Study

Due to the SMS results, we designed an interview-based field study with IT industry professionals. Our intent was to better understand the practical relation of the DS and PM areas, what are today's challenges of projects involving DS, as well as what could be used as possible countermeasures for such challenges.

Semi-structured interviews were conducted with 16 professionals from Brazil (14) and United States (2). The selection criteria was based on their IT industry experience (at least 10 years) and ability to be critical (as perceived by the researchers).

Their role distribution was: 9 project managers, 2 development leaders, 2 test leaders, 1 business analyst, 1 architect, and 1 IT Manager. In average, they had: 17.2 years of work experience, 12.5 years of technical work experience, 6.7 years of managerial experience, and 5.8 years of experience with the current exployer. Next, we briefly present the findings of our field-based study.

2.2.1 Technical Project Managers

The perception of 68.75% of our interviewees is that project managers usually are not involved with technical aspects in the projects they manage. Still, 62.5% considered beneficial, project delivery wise, to have project managers with technical knowledge.

2.2.2 Awareness of System Distribution

Regarding awareness of what DS is, 62.5% of the interviewees were not even familiar with the concept. After Section 1 definition was provided, all interviewees confirmed they now understood the concept, having 84% of them claimed to have participated in DS projects in the past 5 years.

Therefore, the high volume of today's software development projects involving DS does make it difficult even for experienced professionals to realize how frequently they are inserted in such context. For them, these are "just regular projects", where DS is a almost a mandatory solution aspect. This constitutes evidence of an "abstraction trend" of the DS feature.

2.2.3 The Challenges from DS Projects

The discussed challenges of DS projects were either technical or managerial aspects of software development. Each interviewee was allowed to provide as many challenges as they wanted, including ones for a same item. The challenges were then grouped into categories.

The list of categorized main technical challenges and their individual occurences is as follows: Testing (14), IT infrastructure (17), integrations (6), fidelity of non-production to production environments (4), system security (3), system architecture (6), requirements (7), deployments (7), existence of too many implementation options (3) and others (8).

We also discussed managerial challenges related to DS projects. The list of categorized main managerial challenges and their individual occurences is as follows: obtain a skilled team (5), risk management (9), knowledge management (5), team management (4), communication (8), vendor management (5), project planning (6) and others (7).

After the interviews, the main definition of "system distribution" of our study was restricted to solutions that are: (i) distributed regarding their IT infrastructure, e.g. a software distributed between an application server and a database server; and (ii) distributed among different softwares, integrated with each other through interfaces or other mechanisms that allow exchanges, such as of data, tokens, etc.

2.2.4 Failed DS Projects

From the DS projects that the interviewees participated in the last 5 years, an average project failure of 38,44% was reported, having 81,25% of the

interviewees claimed, based solely on their perceptions, to see failure reasons that could be linked to the system distribution aspect and the aforementioned technical and managerial challenges.

The interviewees then provided a set of countermeasure choices they would like to have for dealing with DS projects challenges. The most recurring one was a Software Development Life Cycle (SDLC) specialized in DS (9 occurrences). The other choices were a development framework (2), PM framework (1), diverse tools (3) and others (1).

According to Taylor (2004), "an SDLC is a subset of the project life cycle", "focused on accomplishing the product requirements". The main difference from Project Life Cycle (PLC) activities is that SDLC activities focus on technical aspects of project deliverables while PLC ones are more related to management and leadership (Taylor, 2004).

3 AN APPROACH FOR DS SDLC

Given the discussed results, we propose an SDLC optimized for running Software Development Projects involving DS. The top-level structure of our SDLC contains its key phases, activities, and deliverables, all adherent to the generic SDLC process defined by Taylor (2004).

Because of this generic nature, our main contribution is thereby on the differentiated practices we are proposing, and that should be used in association with the organized structure of the SDLC. These practices are adaptations on well-known and disseminated items, such as a Project Architecture Document or a System Requirements Document for example, tailoring them for use within DS Projects.

Our Phase-Activity-Deliverable structure has to be viewed then as a non-prescriptionary guide. It can be used as is, but it also is easily mappable against different SDLC versions in use by IT companies around the world, which means they could keep using their own processes while simply adding our proposals to them, as they see fit.

3.1 Overall View of Our DS SDLC

The proposed SDLC is designed for an optimal implementation with Waterfall (Pressman, 2001). Adaptations are also proposed for use with Scrum, since one cannot ignore its growing use in today's industry, as demonstrated by VersionOne (2015).

We present our DS SDLC and its associated practices next, all in high-level detail due to space restrictions. We have represented the activitiy flows of each phase through activity diagrams, compliant with Unified Modeling Language (UML) notations. One customization to the notations was made, related to the representation in the diagrams of inputs and outputs for each activity. Inputs are represented on top left and outputs on bottom right of each activity.

Our suggested SDLC practices and some examples related to them are textually described. For the software integration proposals (type of solution 'ii' as defined in Section 2.2.3), examples are based on software integrated data-wise (exchange of data through data interfaces, for example).

3.1.1 Vision Phase

In the Vision Phase, the Project is initiated through the assignment of a Project Manager and initial project team. Preliminary project planning is made by obtaining high-level time and cost estimates. Visibility on DS Project challenges should exist, so that due countermeasures can be planned and implemented, as early as possible in the project. An overall view of the Vision Phase activities, with its inputs and outputs can be seen in Figure 1.

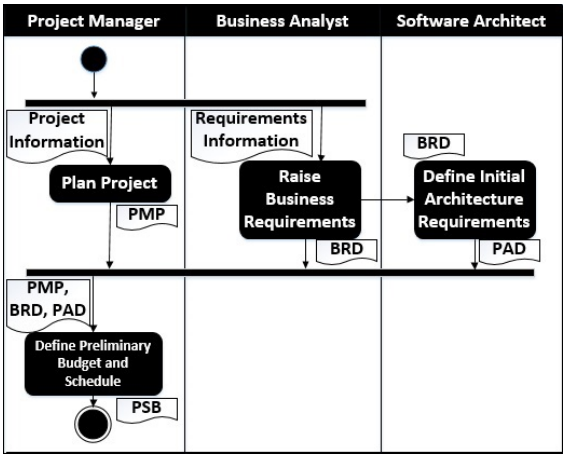


Figure 1: Vision phase of proposed Waterfall cycle.

Our Recommended Practices for Waterfall

- Business Requirements Document (BRD) should have a section for "Business Integrations", which is filled in with key details of all identifiable integrations at a business level (business process, data flow, integration class, etc.);
- DS Non-Functional Requirements (NFR) should be documented in the BRD for the application being developed and for each of its integrations (level of availability needed, number of simultaneous connections, data volume and data periodicity, etc);

- Project Architecture Document (PAD) should have a DS Section, containing visual, incremental architecture information. Business requirements are mapped against integration / infrastructure requirements;
- PAD should include applicable integration/infrastructure technical information, such as data format, data contract, security measures, error handling and logging etc;
- Due care is provided for Project Management Plan (PMP) auxiliary plans, such as Stakeholder and Communication Plans. A customer, a technical and a management liaisons are appointed for each integration;
- Risk Register (RR) should start with a default list of DS risks. The list is continually refined by Project Management Office (PMO) through feedback coming from live projects. It becomes available for upcoming projects;
- All documents above are potential inputs for the Project Schedule and Budget (PSB);
- PMP, BRD and PSB are baselined.

How These Practices Support Waterfall?

- Provision of visibility around integrations / infrastructure demands, as early as possible;
- Better stakeholder identification, reducing the chances of late engagement and Change Requests;
- Helps all stakeholders in setting up their new mindset about the true complexities of their project, as early as possible;
- Schedule and budget are more realistic, as the distribution characteristic is now considered.

Adapting These Practices for Scrum

- Product Owner identifies integration needs before Project start;
- Integration / infrastructure needs are discussed during the first project meeting, usually the Release Planning one;
- Infrastructure and support teams are encouraged to be on-board the discussion already in this phase, early in the project;
- Integration / infrastructure requirements should be treated as user stories, added to the Product Backlog and prioritized according to their value;
- Definition Of Ready (DOR) should take in consideration the DS characteristics of the project in question. For example, it could include "complete data contract being available" and/or "data sample being available".

How These Practices Support Scrum?

- "All" project aspects really become visible to everyone at all times, including the ones related to system distribution, which tended to be "suppressed" before;
- Delivered functionalities will tend to be more stable, as DS key characteristics receive proper attention. Value delivered and perceived increase.

3.1.2 Planning Phase

In this Phase, one executes final project planning based on information now available. Full architecture and infrastructure requirements assessments are now possible, and any gaps before execution should be addressed. System design is complete. An overall view of the Planning activities, with its inputs and outputs can be seen in Figure 2.

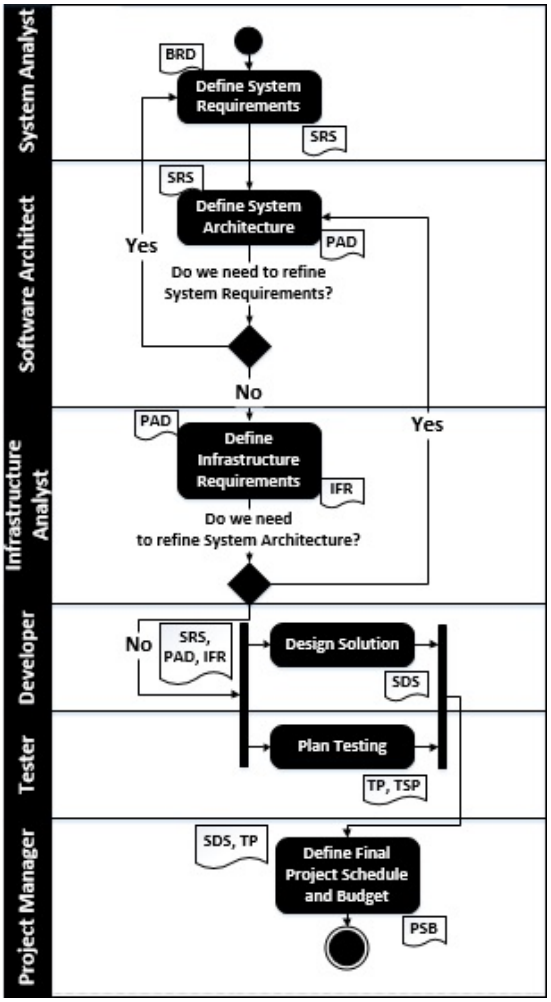


Figure 2: Planning phase of proposed Waterfall cycle.

Our Recommended Practices for Waterfall

- PAD registers the detailed business process flows that will support the solution;

- The Infrastructure Document (IFR) should register detailed information on required infrastructure. Hardware, software and networking needs are mapped, especially the ones affecting system distribution, such as servers' latencies and locations, ports and protocols, etc;
- RR is updated with new risks, including DS ones;
- System Requirement Specification (SRS) must be created and kept in close alignment with BRD and PAD, thus making sure no previoulsy raised system distribution key definitons are lost. These should instead only be incremented in the SRS, thus making the work to create this artifact easier;
- Test Plan (TP) must include detailed information about the needed environments, data masses, log testing etc. It also could include the plan for test environment redundancy, in case part of tests are in the Project's critical path;
- Test specification must be created and kept in close alignment with SRS, thus making sure no previoulsy raised system distribution key definitons are lost;
- System Design Specification (SDS) must be created and kept in close alignment with the SRS, thus making sure no previoulsy raised system distribution key definitons are lost. They should instead only be incremented in the SDS;
- SRS, PMP and PSB are updated and re-baselined.

How These Practices Support Waterfall?

- End of Planning phase has all major solution specifications and a complete design, all considering the DS characteristics of the project;
- Improved visibility acquired regarding what are the main technical constraints and risks for the rest of the project, before execution.

Adapting These Practices for Scrum

- There should be acceptance criterion created for each infrastructure / integration story, such as:
  - What should be the systems' behavior when the integrations are and are not available?
  - What should be the systems' behavior when the data contract is or is not being respected, regarding for example, data consumption and data transformation?
- Integration / infrastructure scope are treated as user stories and are added into a Sprint Planning scope, if Ready criteria is met.

How These Practices Support Scrum?

- Clear prioritization of Integration and Infrastructure aspects in relation to regular software requirements, all based on their now perceived value for the solution;
- Raised DS acceptance criterion will later be used during development and testing cycles. Due importance is provided to the validation of the system distribution key characteristic.

3.1.3 Building Phase

In the Building Phase, one assembles the required Non-Production and Production infrastructures, as well as creates the software product through codification and developer´s level testing. Finished Test Cases are also an output of this phase. An overall view of the Building phase activities, with its inputs and outputs can be seen in Figure 3.

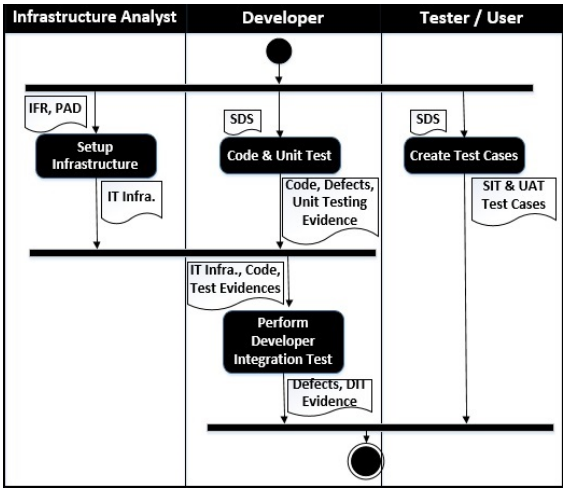


Figure 3: Building phase of proposed Waterfall cycle.

Our Recommended Practices for Waterfall

- IT Infrastructure, non-production (and production if possible), is raised during this phase. Attention to all needs mapped in the infrastructure document is essential;
- Logging and monitoring functionalities must be implemented according to the strategy previously mapped in the PAD. This allows easier traceability of defects in non-production environment, as it will be possible to quickly identify from which application the defect comes. Also, when in production, traceability of incidents will also be benefited by the same approach;
- Developer Integration Test (DIT) first includes only mocked integrations, but in a second moment, if possible, will be done with all integrations in the non-production environment, thus simulating what will be found in production.

How These Practices Support Waterfall?

- Completion of the development step with a much more stable code, mainly due to attention given to key distribution details.

Adapting These Practices for Scrum

- Sprint Zero includes the assembly of non-production infrastructure;
- Sprint Zero includes test analysis for test scenarios generation. Next sprints have the same approach. This generates better coverage during test execution.

How These Practices Support Scrum?

- System analysis team is one step ahead of the rest of the team, thus making sure requirements are well understood before actual implementation. Same happens to test team, and now the project benefits from the "planned in advance" testing.

3.1.4 Testing Phase

In this phase one performs detailed integrated testing from both the test team's and user's perspectives. Defect management and handling happen during the entire phase. Performance testing, when applicable, is also carried out on this phase. An overall view of the Testing Phase activities, with its inputs and outputs can be seen in Figure 4.

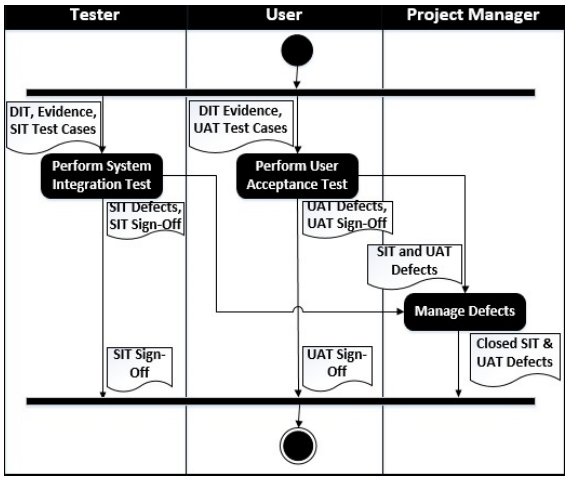


Figure 4: Testing phase of proposed Waterfall cycle.

Our Recommended Practices for Waterfall

- Testing should provide an important focus on the Non-Functional Requirements (NFRs), considering they highly influence the system distribution decisions;
- Mocked data should be avoided at this stage. The use of data masses that are the closest possible to production is encouraged;
- Mocked integrations should be avoided at this stage. It is ideal to have all systems integrated in the testing non-production environment;
- Test infrastructure and overall environment in use must be the closest possible to production;
- Sign-offs should be received from who is performing the tests by the end of System Integration Testing (SIT) and User Acceptance Testing (UAT).

How These Practices Support Waterfall?

- An independent test team will validate what was built and delivered;
- Stabilization of defects prior to handing the system over to users for UAT testing;
- Realistic testing will help in preventing many incidents in production.

Adapting These Practices for Scrum

- Production environment can be raised and be continuously refined at this point;
- Proposed waterfall test practices can be used equally in Scrum, without adaptations.

How These Practices Support Scrum?

- The benefits are the same coming from the proposed practices in Waterfall Testing phase.

3.1.5 Releasing Phase

In this phase one provides the support team and users with the application training. Application is made available for use in production. Provision of warranty for the application, through the solution of production incidents. Project closure is executed. An overall view of the Releasing Phase activities, with its inputs and outputs can be seen in Figure 5.

Our Recommended Practices for Waterfall

- A deployment and rollback plans should be available for tracking of all the deployment tasks and their impacts to each integration;
- A post-deployment plan should be available in order to help validating if all core functionalities from the deployed / integrated systems are unnafected and available;
- A "System Profile" Document (SPD) describes, in business terms, the implemented system, its purpose, integration points, data flowing in and out, etc. This is the base of the Knowledge Transfer (KT) for the support team and users;

- Lessons learned document captures learned items that will be inputs to upcoming projects. A DS section exists in the document.

How These Practices Support Waterfall?

- Project closure occurs when the system is fully transitioned to production and accepted by users;
- System transition to the support team is also needed for closing out the project.

Adapting These Practices for Scrum

- If there is not enough time in last sprint, then create a "Sprint-F" (for Final), for carrying out KT and the remaining documentation, including SPD;
- Lessons learned are filled out as part of final Sprint Review and Sprint Restrospective, using Sprint-F for that as well, if needed.

How These Practices Support Scrum?

- Documentation is generated only until it generates value for the users / customers;
- Project closure happens when expected product value has been delivered;
- System maintenance is considered, as there is the foment of KT for that purpose;
- Continuous improvement of projects through the raise of lessons learned.

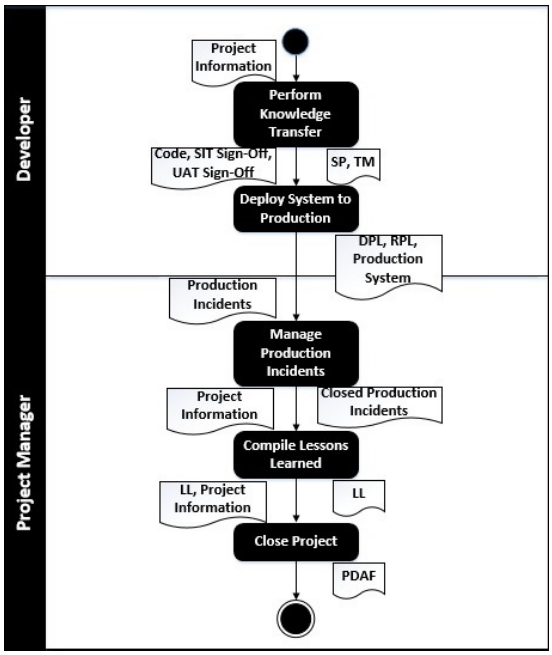


Figure 5: Releasing phase of proposed Waterfall cycle.

3.1.6 Monitoring and Controlling Phase

This phase happens in parallel to the project, providing oversight for all phases. Change impacts are monitored, action being taken when needed and status being reported. An overall view of the Monitoring and Controlling (M&C) Phase activities, with its inputs and outputs can be seen in Figure 6.

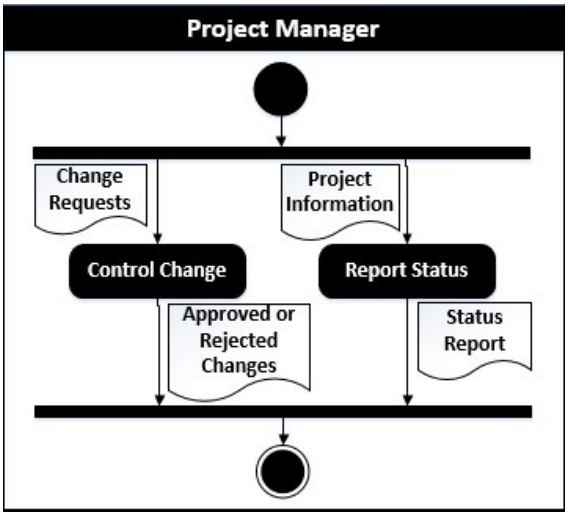


Figure 6: M&C phase of proposed Waterfall cycle.

Our Recommended Practices for Waterfall

- Status reports addresses the DS aspect, describing what is the status on infrastructure as well as on each integration. Items such as difficulties faced, steps completed, opportunities, risks, teams engagement, etc should be on the report.

How These Practices Support Waterfall?

- Synchronization of all stakeholders' visibility on all key project aspects, including the DS one;
- Foment of the whole team's participation on all project issues and decisions.

Adapting These Practices for Scrum

- Daily Scrums, Sprint Plannings and Release Plannings may have part of their time dedicated for the review of the teams' accomplishments regarding infrastructure / integrations items.

How These Practices Support Scrum?

- "All" project aspects become visible to everyone, including the ones related to DS.

3.2 Validation and Limitations

We define this research as an empirical, qualitative one. As such, our DS SDLC and practices, after created, went through the process of "*member checking*", a traditional validation technique used in

empirical work (Singer, 2007).

We invited 5 participants from the 16 IT professionals, who had previously participated on our interview-based field study, to participate again in the validation session. They were chosen due to the authors' perception of their highly critical opinions as well as the importance of their previous contibutions.

We also invited 2 additional professionals that had no previous contact whatsoever with this research. They were selected based on their seniority as IT professionals, each one having more than 15 years of work experience in IT. Both were from Brazil.

The feedback obtained was encouraging. These professionals all agreed that many practical benefits should come from the implementation of our proposed SDLC and practices. All of them also had their own inputs with improvements, which in turn led to the version of our model discussed in this paper.

We did not include in preliminary validation the application of the SDLC in real-life projects given time constraints. However, the field study provided us with an initial rich data set that suffices before we continue our work. Our next step is, then, to observe how the SDLC is welcomed in real-life projects and what suggestions industry professionals will make to improve and further its scope, if any. For now, the current limitations of our study are as follows:

- No practical experiments with real projects and/or companies conducted so far;
- Field study participants were from Brazil and the United States only while member checking participants were from Brazil only;
- Small diversity of companies, (one American company providing 9 out of 16 participants);
- Our SDLC currently does not drill down to task-step structure;
- The SDLC has some generic management activities and deliverables. These will migrate into an independent PLC in the future.

4 CONCLUSIONS

In this work we discussed the many challenges brought by DS to Software Development Projects. Little research exists though on the intersection of Distributed Systems and Project Management.

As presented in our results, professionals from the IT industry do recognize the importance of understanding those challenges and taking systematic actions in order to mitigate or eliminate most of them.

We believe that our SDLC and related practices are in line with the industry needs for an effective countermeasure for the identified challenges, addressing them by broadening project teams´ awareness about the importance of properly handling the System Distribution aspect on the projects they are inserted in.

The SDLC will also provide elements to facilitate communication with users and customers, allowing them to realize how complex a software truly is, not only from a regular requirements perspective, but from technical and infrastructure perspectives as well.

More research is still needed for verifying the effectiveness of our proposals, as well as their easiness of use, both when used by themselves as well as when simply coupled to other SDLCs. This is the cornerstone of our research's next step.

REFERENCES

PMI Inc., 2015. What is Project Management?. Consulted on October 13, 2015. Available at http://www.pmi.org /about-us/about-us-what-is-project-management.aspx.

PMI, 2013. A Guide to the Project Management Body of Knowledge: PMBoK Guide. *Project Management Institute. Pennsylvania, 5th edition*.

Coulouris, G., Dollimore, J., Kindberg, T., Blair, G., 2012. *Distributed Systems Concepts and Design, Addison-Wesley, Boston, 5th edition*.

Taylor, J., 2004. Managing Information Technology Projects: Applying Project Management Strategies to Software, Hardware and Integration Initiatives. AMACOM. New York.

Version One, 2015. "*The 9th Annual State of Agile Report*". Consulted on October 07, 2015. Available at http://info.versionone.com/state-of-agile-development-survey-ninth.html.

Herbsleb, J. D., Moitra, D. 2001. Global Software Development. *In IEEE Software, V.16, n.2, 16-20*. IEEE.

Pressman, R, Maxim, Bruce, 2014. Software Engineering A Practitioner's Approach. *McGraw-Hill. New York, 8th edition*.

Shull, F., Singer, J., Sjoberg, Dag, 2007. Guide to Advanced Empirical Software Engineering. *Springer. New Jersey, 2008 edition*.

The Concept of Project Management Platform using BI and Big Data Technology

Jolanta Pondel[1] and Maciej Pondel[2]
[1]*University of Business in Wroclaw, Ostrowskiego 22, 53-238, Wroclaw, Poland*
[2]*Wrocław University of Economics, Komandorska 118/120, 53-345, Wrocław, Poland*
jolanta.pondel@handlowa.eu, maciej.pondel@ue.wroc.pl

Keywords: Project Management, Project Management Software, Workflow, Big Data, Business Intelligence.

Abstract: In current world, organizations need to adapt to the changing business environment. They decide to conduct projects that result with new business processes, new products or services. Very often the goal of the project is to streamline specific area of company or a whole business. The projects become a very complex set of activities that require a sophisticated IT tools to support the efficiency of all the actions. Probably none single software application is able to handle every aspect of the project. That is why the authors decided to identify the kinds of software necessary for supporting the project and choose the applications that from their perspective can aid specific project activities. We have to remember that projects can generate a significant amount of data. If we are able to transform data into relevant information, we can maximize the possibility of success (both project and organization). In this paper authors propose the foundation of a complex platform supporting project management and execution with an emphasis on the analytical and reporting part by usage of Business Intelligence and Big Data technologies. Evaluation of such a platform is the subject for the future work.

1 INTRODUCTION

Nowadays organizations are facing new challenges. They run multiple project simultaneously to achieve various business goals. They also gain the experience from completed project that finished with a success, partial success or a failure. Enterprises require software applications to support every aspect of project management and execution. All those applications generate a significant amount of data. The data is stored in various IT systems in miscellaneous formats and very often in different locations. The effectiveness of the project decision-making is not resulting only from the amount of data collected, but also depends on the ability to the proper choice of sources of information. The speed of extracting the information is also crucial if we want to make the most successful decisions and limit the risks appearing in projects or regular business activities. The usage of advanced tools supporting the project management and reporting application very often based on the artificial intelligence seems essential to run a business, particularly in area of a project management.

2 PROJECT MANAGEMENT

Organizations decide to run projects when they want to (see (Burke, 2013)):

- deliver products or services to outside customers,
- increase internal efficiency by introducing the internal change.

Projects are activities in companies that have little (usually none) repeatability but a very high degree of complexity. Typically, those undertaken actions are related to the new (unique) activities that bring solutions for a new business situation or a problem. To achieve the effect, we must specify, among others, its duration and costs. We usually assign the author / owner / manager for those actions who is responsible for achieving the final result of the project.

The international organization consisting of companies and individuals interested in managing projects - Project Management Institute defines a project as a temporary activity that is undertaken to provide a unique product / service or achieve unique results (Kerzner, 2013). Z. Szyjewski believes that the project is a unique, non-routine process meeting specific targets in a given time by means of specific

Pondel, J. and Pondel, M.
The Concept of Project Management Platform using BI and Big Data Technology.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 166-173
ISBN: 978-989-758-187-8

measures (Szyjewski, 2004) (see (Pondel and Pondel, 2011)).

Project can be also defined by indicating its individual characteristics. Various authors claim that a project is non-repetitive, time-limited and it has defined objectives. It includes various management methods and techniques. It solves new and previously unknown problems and it is associated with certain risks. Project must have a corresponding budget and during the performance of work, the project participants are under pressure (Kellner, 2001).

The basic attributes of the project include: location in time, uniqueness, complexity, purposefulness.

Project management can be defined as a set of managerial activities related to the implementation of projects and a set of used in these operations principles, methods and tools (Guide, 2001). Project management involves the application of knowledge, experience, tools, methods and techniques during the project activities, to achieve or even surpass the needs and expectations of stakeholders. Implementation of the project requires meeting many aspects, such as: scope, time and quality, various needs and expectations of stakeholders, identified and anticipated requirements, risks and their neutralisation plans.

Modern organizations to streamline their operations and project management, use the access to various electronic information resources. Multitude of available information and the diversity of sources make the decision-making more complex. We should take into account such factors as: reduction / extension / asymmetry of time and information and the responsibility of many people for making decisions (various locations of the company). At each stage of project in companies we can identify many of the key elements that influence the success of the whole project execution. All this encourages companies to investigate and use different types of IT tools that allow to facilitate efficient decision-making process.

Managing the project, we have to be aware that it requires the efficient communication and proper relationships management. Those relationships exist (Kerzner, 2013):

- within the project team,
- between the project team and the functional organizations,
- between the project team and senior management,
- between the project team and the customer's organization, whether an internal or external organization.

3 SOFTWARE SUPPORTING PROJECT MANAGEMENT

According to many sources we can divide the software supporting project management into the following groups (see (Rus and Lindvall, 2002), (Wikipedia, 2015)):

- Collaborative software,
- Issue tracking system (ITS),
- Planning / Scheduling,
- Project Portfolio Management,
- Resource Management,
- Document Management,
- Workflow system,
- Reporting and Analyses.

Team collaboration is essential for the success of projects. When team members are spread across different locations, individual awareness of the activity of others drops due to communication barriers (Hattori, Lanza, 2010).

Collaboration software is designed to improve productivity of individuals, teams and organizations. This is achieved through the following capabilities of collaboration software (see (Hildenbrand and Rothlauf and Geisser and Heinzl and Kude, 2008)):

- informing,
- coordinating,
- actually collaborating,
- cooperating.

Issues are common part of every project. They may appear on every stage and requires the actions leading to its successful resolution.

An **issue tracking system (ITS)** is a software application that allows an enterprise to record and follow the progress of every problem or "issue" that a team member identifies until the problem is resolved. With an ITS, an "issue", which can be anything from a simple customer question to a detailed technical report of an error or bug, can be tracked by priority status, owner, or some other customized criteria.

An ITS provides the user with a way to report an issue, track progression towards its resolution, and know who is responsible for resolving the issue. It also allows the manager of the system to customize the tracking procedure so that unnecessary documentation on the part of the problem solvers does not become a waste of time. Many kinds of enterprises use ITS applications, including software developers, manufacturers, IT help desks, and other service providers (Techtarget, 2015).

Planning is determining what is necessary to be done, who should be responsible for the task, and when the task should be completed to fulfil defined

requirements. We have to consider the following element of planning (see (Kerzner, 2013)):

- Objective – a goal to be achieved.
- Schedule – a plan defining in what point in time the activities will be started and when they will be completed. It shows also the resources assigned to the task and people responsible for task successful execution. In the schedule the references and dependencies between activities must be also presented.
- Budget – planned expenditures required to achieve objectives.
- Forecast – a projection of what will happen in a certain moment in time.
- Organization – a list of position of team members with corresponding duties and responsibilities required to complete defined tasks.
- Standard – a level of individual or group performance defined as adequate or acceptable.

We have to be aware that planning is based on forecasting and the uncertainty is involved with planning in an inseparable way. That is why planning is a continuous process of making decisions and organizing the effort needed to carry out these decisions. Planning must be based on monitoring the completed tasks and designing the future in order to achieve goals. If the systematic planning is not effected, it ends up with reactive management leading to crisis management, conflict management and firefighting.

Software supporting **planning and scheduling** often use a project structure to describe a given project. A project structure maps real-world aspects of a project, such as timelines and tasks, into an electronically accessible format. For example, many project development systems describe a start, finish, and other schedule dates of a project, the tasks that are performed and the results that are achieved during the project, and the data objects that are generated by the project or used to complete the project. A Gantt Chart is an example of a project structure that can be used to describe a given project. A Gantt Chart is a graphical representation that shows the time dependency of several tasks of a project within a calendar. A Gantt Chart provides a graphical illustration of a schedule that helps to plan, coordinate, and track specific tasks in a project (Meyringer, 2006). Gantt Chart is most commonly used in a software supporting project planning.

The ultimate goal of **Project Portfolio Management** is to maximize the contribution of projects to corporate success. Thus, PPM can be considered as the simultaneous management of the collection of projects that make up an investment strategy of a company (Heising, 2012). Project Portfolio Management is about more than running multiple projects. Each portfolio of projects needs to be assessed by its business value and adherence to strategy. The portfolio should be designed to achieve a defined business objective or benefit. Project management guru Bob Buttrick summarised it when he said; Directing the individual project correctly will ensure it is done right. Directing 'all the projects' successfully will ensure we are doing the right projects (Projectsmart, 2015).

The most important features of Project Portfolio Management Software are:

- project evaluation process or methodology,
- cost and benefits measurement,
- progress reporting,
- communication of key project data, for example executive dashboard,
- resource and capacity planning,
- cost and benefits tracking.

Resource management software is supporting users in following tasks (see (Kerzner, 2013)):

- Resource levelling is an attempt to avoid the manpower peaks and valleys by smoothing out the period-to-period resource requirements.
- Resource allocation which is an attempt to find the shortest possible critical path based upon the available resources.

During every project execution a number of documents appear. Document management systems are essential to store, share, search and protect the documents. Some of the key features in document management include:

- Check-in/check-out and locking, to coordinate the simultaneous editing of a document so one person's changes don't overwrite another's.
- Version control, so tabs can be kept on how the current document came to be, and how it differs from the versions that came before.
- Roll-back, to "activate" a prior version in case of an error or premature release.
- Audit trail, to permit the reconstruction of who did what to a document during the course of its life in the system.
- Annotation and Stamps.

Workflow systems are considered mainly as tools supporting business processes. A workflow application implements a business process model. The model describes the process steps to be performed to achieve a specific business goal, business rules for coordination of those steps and responsibilities of process participants (Schmidt, 1998). The steps include tasks that should be

performed by agents that can be human, computer systems or combination of both (Demeyer, 2010). Workflow systems, with the benefits of efficient and flexible process modelling and process automation, have been widely used for managing business processes. Although the business process and project are two different subjects (business process is repetitive and project goal is always to create the individual deliverable) the stages or tasks in projects can be treated as a small process that should be executed according to the business rules defined in a workflow tool eg:

- Document approvals - business rules define who is responsible for creation and approval of documents. Every kind of document can have individual list of approvers.
- Change management – the workflow can define how the change should be identified, described, estimated and who should be responsible for its approval and execution.
- Risk management – the workflow can enforce the specified risk description by a project manager and can lead the process of execution of preventive actions.
- And many more.

Reporting and analyses are essential when we would like to control and monitor all aspects of the project execution. We can rely on a reporting modules of mentioned software to prepare simple analysis (usually as tables or charts) presenting the information from one area of project management and execution field. We can also use Business Intelligence tools that could integrate the data from all the systems used during project and present the holistic reports. Regarding Business Intelligence tools we can distinguish 2 main approaches:

- traditional BI based on ETL Process, data warehouses, data marts, OLAP, dashboards, scorecards and analytics,
- Self Sevice BI where Power Users connect to various data sources and create their data models on which they build visualisation layer.

Authors believe that for more sophisticated purposes also the techniques called Big Data can be useful in a project management.

4 THE CONCEPT OF SOFTWARE PLATFORM SUPPORTING PROJECT MANAGEMENT

As it was mentioned in the previous chapter we can distinguish several roles that PM software can play and there is a number of software applications between which we can choose the most efficient and convenient tools.

Depending on the project specifics, we can define different criteria of PM tools selection. For a purpose of this paper we will take the following assumptions:

- we will focus on IT projects,
- a platform must support not only individual project but a number of projects that are conducted in the organization,
- a majority of project team members are office workers, but we can meet also handworkers dealing with hardware installation, computer network construction, inventory delivery,
- a significant portion of project members and stakeholders are mobile workers who travel a lot and use mobile devices for professional purposes.

Taking into account those conditions authors will try to choose the list of IT systems that will meet the following criteria:

- They have an open API to allow integration with other items of the platform.
- They are portal solutions – allow access through the Internet Browser.
- They are can be hosted in cloud environment.
- They should provide the mobile access to their features.
- They should support the world wide standards (eg. most common files formats, ways of data presentations).

The proposed solution is aimed to be a comprehensive platform that can support every single aspect of project management and execution.

While choosing the software tools authors followed previously defined criteria, their own experience, popularity of software tools, ability to integration with previously selected tool and available description of chosen tools. Authors do not claim that every chosen software product is the best in its category. For sure the discussion about better selection of tools could be initiated.

For collaboration and document management platform authors chose the services being a part of Microsoft Office 365 Platform. Those are cloud services that contain: Yammer – the world leader of social software, MS SharePoint – the platform for document management, MS Exchange that provide the features for business email, calendars and task management, Skype for business that is unified communication platform providing such features as: IM, audio and video calls, online meetings and sharing. Authors decided to use those software tools, because they are compatible with MS Office which is

the most common tool for document creation. This platform is also considered as a world leader in Social Software (Gartner 2015). It also includes a number of features that together constitute the unified platform for collaboration, communication, information management and document management. It is possible that we could find in every single area some specified product that could be in some criteria better that those chosen, but in would require integration with the rest of tools. In case of Office 365 those tools are already integrated.

Table 1: The list of software tools constituting the holistic Project Management Platform.

Type of software	Chosen IT system
Collaborative software	Yammer, MS SharePoint Online available in MS Qffice 365, MS Exchange online, Skype for Business
Document Management	
Workflow system	MS SharePoint Online with Nintex Workflow and Nintex Forms for Office 365
Issue tracking system (ITS)	Atlassian Jira
Scheduling	MS Project
Project Portfolio Management	MS Project Online
Resource Management	
Reporting and Analyses	Data Warehouse: MS SQL Server Business Intelligence, BI / Self Service BI: QlikView, MS PowerBI Big Data: Hadoop MongoDB Pentahoo Business Analytics

Authors decided to build the workflow platform also on SharePoint to keep the consistency of tools. Microsoft platform contains the Workflow engine available to SharePoint. Unfortunately, in its original form it is difficult to be applied so authors chose the application for modelling and maintaining the processes called Nintex for Office 365. That include the tool for process automation (Nintex Workflow) and a forms designer application (Nintex Forms).

For issue tracking and task management in project authors chose the Atlassian Jira Software that allow:

- Planning tasks and assigning them to project members.
- Tracking the work of team members.
- Collaboration and communication in terms of assigned tasks and issues.
- Creating workflows automating tasks and issues execution.

Jira was chosen because its large functionality and existence in many rankings on top positions eg. Gartner considers Atlassian products as one of the leders in his Magic Quadrant for Application Development Life Cycle Management together with IBM and Microsoft Products (Atlassian, 2015). We must add that Jira is used not only in software development projects but also in many more types of projects.

We can observe that some features in Jira exist also in Microsoft Office 365 Platform. Authors assume that the collaboration and information management on a management level will be performed in the Microsoft Office 365 platform. The task management on a project execution level will be performed in Jira. Moreover, in the specific areas those platforms must be integrated to provide a consistent tool useful for both managers and project team members.

Regarding the Scheduling on a managerial level and also project portfolio management authors propose to use the Microsoft platform that consist: MS Project Professional Application for scheduling purposes and MS Project Online which is an EPM (Enterprise Project Management) tool allowing the management of whole Project Portfolio. Together with portfolio management this platform includes the resource management capabilities. It is directed to project managers, project stakeholders and the management personnel involved in the project. This platform requires integration with task and issue tracking system (Jira) which is directed to project executors. The integration has the following aims:

- Convey the information about scheduled actions to Jira and assign specific tasks to the team members.
- Inform back the Project Management Platform about a current state of assigned tasks.

The diagram visualizing the concept of the platform is presented on the Figure 1. It doesn't include the reporting and analytical platform that will be described in the next chapter.

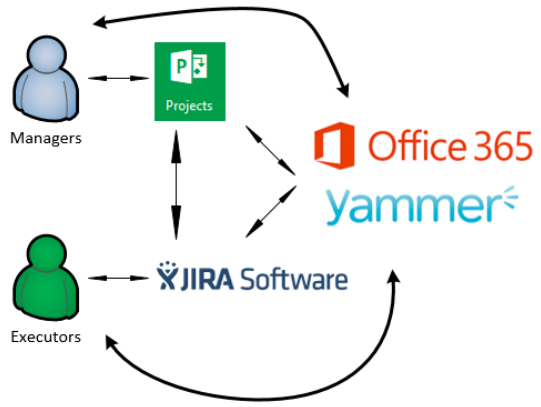


Figure 1: The concept of Project Management Platform.

4.1 Reporting and Analyses

In a project management processes and portfolio management decision making is an immanent activity of managers, project owners, stakeholders and sometimes also project executors. That is why analytical platform can have a crucial meaning in making decision regarding Project and Project Portfolio. Those decisions regard among others:

- scheduling,
- resources utilisation,
- risk management,
- approvals,
- technological decisions.

Fundamental assumption of an analytical system in a Project Management Platform should be provision of targeted information for every layer of its users. That is why we propose to build the analytical tool based on a 3 pillars:

- Data warehouse with a regular BI system,
- Self-Service BI platform,
- Big Data platform.

Authors decided to base a data warehouse on a Microsoft SQL Server capabilities that can used as on-premises solution and also the cloud service hosted in MS Azure can be used. The presentation layer for the Business Intelligence system can be the MS Power BI application that is part of Office 365 so it is consistent with the other components of platform. However, authors recommend using also other ways of information presentation like QlikView which is the leading tool for data analytics and visualisation.

In the BI platform we would gather data from every component of our Project Management Platform and allow to analyse the following characteristics:

- Project Portfolio Management - Data regarding project characteristics, timelines, objectives and deliverables.
- Scheduling - Data describing the timelines and the progress of the project and influence of the materialised risks on the project schedule. Also the changes in project timelines. They include also financial data and the project efficiency.
- Resource management - The estimations and real resources utilisation. The resource characteristics. The references between resource skills and their efficiency.
- Document Management - Document metadata (dates of creation, authors, dates of modifications, etc.).
- Issues and Tasks Tracking - The amount of issues and tasks at specific stage of the project, the resource consumption during tasks execution and issues solving, the types of issues.
- Workflow - The current progress of every process, planned dates of process completion.
- Collaboration support - The number of topics discussed during project planning and execution.

The analysis available on this layer would be directed mainly to the portfolio managers, project managers and whole management personnel. Sometimes the specific analysis describing specific project of specific resource efficiency would be useful for the team members.

Self-Service BI platform would be directed mainly to project managers. As mentioned earlier the aim of the project is to deliver a unique product or a service that is why every project has its own individual specifics and characteristics. Looking from this perspective we should be aware that it may be impossible to build a universal data warehouse that can cover every specific information requirement. That is why the BI tool that enables creation of specific and individual reports would be very useful in such case. It can be based on the same tools mentioned above.

Regarding the Big Data platform, it can bring the benefits mainly to project managers and team members. We assume that Big Data mechanisms can store mainly the information about all events in the Project Management Platform which can be:

- The statistics about accesses of every team member to every component of the platform. Such analysis can confirm if the project executors possess the sufficient information about the project characteristics, decisions, assumptions and boundaries.

- The changes in documentation and the influence of document lifecycle on the project.
- The issues descriptions and comments of employees providing resolutions can give us knowledge helping risk management.
- Data describing events appearing during process execution. Natural language comments analysis may give us valuable knowledge.
- Media appearing during collaboration, text content of discussions, findings, commitments, conclusions and their influence on the project execution.

Those events should be gathered by event hub mechanism and feed Dig Data repository and trigger user notification if applicable. As described the Big Data platform store mainly unstructured data and we can expect that the amount of the data can exceed the abilities of relational databases to efficient processing (especially in the organisation where a number of projects are executed at the same time and there is a significant number of historical projects that also consist a valuable data). Analysis of events happening in historical project together with the findings and observations relating to the corresponding projects bounds to bring managers the valuable knowledge allowing:

- streamlining the projects efficiency,
- avoiding or minimalizing the risks,
- improving the quality of deliverables.

Being aware of the assumptions and expectations directed to the Big Data platform authors propose to build it using the common technologies like:
Hadoop - framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage (Hadoop, 2015). In our case Hadoop will improve the performance of the system storing a large dataset from a number of projects.
MongoDB - it is an open-source, document database designed for ease of development and scaling. It is one of the most popular and appreciated NoSQL Databases management system and it is positioned by Gartner Magic Quadrant as a Challenger (Mongodb, 2015). MongoDB is equipped with MongoDB Connector for Hadoop what allows to pull MongoDB data into Hadoop Map-Reduce jobs, process the data and return results back to a MongoDB collection.
Pentaho Big Data analytics tools allow to extract, prepare and blend the data. It includes the visualizations and analytics capabilities. It contains: data ingestion manipulation integration, enterprise and ad hoc reporting, Data Discovery and visualisation and predictive analysis. Pentaho Big Data is capable to communicate directly with MongoDB database.

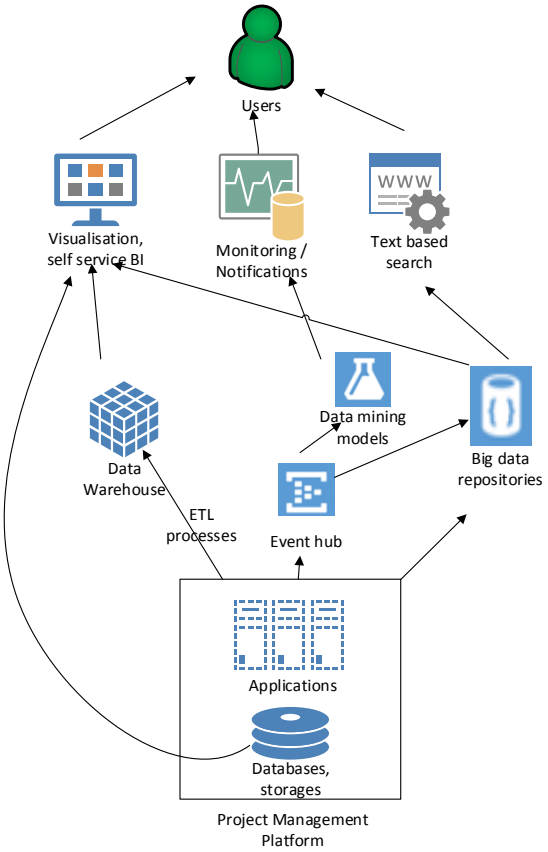


Figure 2: The concept of Project Management Platform.

Presented on a Figure 2 concept of BI / Big Data analysis is a general level considerations of authors and needs to be verified and evaluated during the next stages of research. The highest benefits that authors predict of such approach are:

- Management information visualisation
- Event analysis related to project team communication, collaboration or documentation lifecycle resulting with real time alerts that warn against possible risks and possible project issues. Those alerts are based on data mining based analysis that recommend undertaking of specified actions to avoid predicted problems that may impact the project's success
- Large text sets analysis allowing to search for sufficient project information across all heterogenic systems and applications

5 CONCLUSIONS

The Project Management processes require adequate software applications that together should act as a seamless platform supporting all the actions that can be undertaken. It is essential those applications to communicate and together bring the value to the final users and project stakeholders. It is also crucial to have experienced people that manage and execute the project. In the modern enterprise environment, where a number of projects are executing at the same time a proper data collection and processing seems also essential. Modern techniques of collecting and processing data can benefit for the decision making during the project especially in areas of risks identification, better resource workload estimations, more adequate planning and information and knowledge sharing. The experience gained during project execution is also helpful for improving efficiency of the future projects. Authors of the paper claim that the Business Intelligence tools and Big Data analysis can provide Project Managers, stakeholder and regular team members with a very valuable information and knowledge. Authors proposed the list of software applications that can support the project management processes with a special emphasis on a reporting and analytical capabilities. The future research will contain identification of more detailed Project Management use cases that can be improved by proposed platform. Authors will also focus on empirical verification of effectiveness of proposed platform. Authors are going to investigate every single item of the platform but also want to focus on the evaluation from the holistic perspective.

REFERENCES

Atlassian, 2015. *A Leader in Gartner's 2015 Magic Quadrant for Application Development Life Cycle Management*, https://www.atlassian.com/gartner/

Burke, R., 2013. *Project management: planning and control techniques*. New Jersey, USA.

Demeyer, R., Van Assche, M., Langevine, L., Vanhoof, W., 2010. *Declarative workflows to efficiently manage flexible and advanced business processes*. In Proceedings of the 12th international ACM SIGPLAN symposium on Principles and practice of declarative programming (pp. 209-218). ACM.

Gartner, 2015. *Magic Quadrant for Social Software in the Workplace*, http://www.gartner.com/technology/reprints.do?id=1-2QJAU20&ct=151027&st=sb

Guide, A., 2001, Project Management Body of Knowledge (PMBOK® GUIDE). In *Project Management Institute*.

Hadoop, 2015. *What Is Apache Hadoop?* https://hadoop.apache.org/

Hattori, L., Lanza, M., 2010. *Syde: a tool for collaborative software development*. In Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering-Volume 2 (pp. 235-238). ACM.

Heising, W., 2012. *The integration of ideation and project portfolio management — A key factor for sustainable success*. International Journal of Project Management, 30(5), 582-595.

Hildenbrand, T., Rothlauf, F., Geisser, M., Heinzl, A., Kude, T., 2008. *Approaches to collaborative software development.* In Complex, Intelligent and Software Intensive Systems. CISIS 2008. International Conference on (pp. 523-528). IEEE.

Kellner, H., 2001. *Die Kunst, IT-Projekte zum Erfolg zu fuehren.* Ziele-Strategien-Teamleistungen, Hanser, Wien.

Kerzner, H. R., 2013. *Project management: a systems approach to planning, scheduling, and controlling*. John Wiley & Sons.

Meyringer, M., 2006. U.S. Patent No. 7,050,056. Washington, DC: U.S. Patent and Trademark Office.

Mongodb, 2015. *Gartner Positions MongoDB as a Challenger on the Magic Quadrant for Operational Database Management Systems* https://www.mongodb.com/blog/post/gartner-positions-mongodb-challenger-magic-quadrant-operational-database-management

Pondel, M., Pondel, J., 2011. *Czynniki powodzenia projektu informatycznego*. Informatyka Ekonomiczna, (20), Wydawnictwo Uniwersytetu Ekonomicznego we Wrocławiu, Wrocław.

Projectsmart, 2015. What is project portfolio management? https://www.projectsmart.co.uk/what-is-project-portfolio-management.php

Rus, I., Lindvall, M., 2002. *Guest editors' introduction: Knowledge management in software engineering*. IEEE software, (3), 26-38.

Schmidt, M. T., 1998. *Building workflow business objects*. In Business Object Design and Implementation II (pp. 64-76). Springer London.

Szyjewski Z, 2004. *Metodyki zarządzanie projektami informatycznymi*, Placet, Warszawa.

Techtarget, 2015. *Issue tracking system (ITS) definition*. http://searchcrm.techtarget.com/definition/issue-tracking-system

Wikipedia, 2015. *Comparison of project management software*
https://en.wikipedia.org/wiki/Comparison_of_project_management_software

Conceptual Mappings to Convert Relational into NoSQL Databases

Myller Claudino de Freitas[1], Damires Yluska Souza[2] and Ana Carolina Salgado[1]
[1]*Center for Informatics, Federal University of Pernambuco, Professor Luis Freire ave, Recife, Brazil*
[2]*Academic Unit of Informatics, Federal Institute of Education, João Pessoa, Brazil*
{mcf, acs}@cin.ufpe.br, damires@ifpb.edu.br

Keywords: Relational Databases, NoSQL Systems, Conceptual Mappings, Data Conversion.

Abstract: Sometimes, data belonging to Relational databases need to be transferred to NoSQL ones. However, the data conversion process between Relational to NoSQL databases is considered as not trivial, since it is necessary to have considerable knowledge about the data models at hand. Regarding the structural heterogeneity underlying this problem, we propose an approach, named as R2NoSQL, which defines conceptual mappings to enhance the data conversion process. In this paper, we present our approach and some implementation and experimental results, which show that, by using the defined conceptual mappings, we obtain a consistent target NoSQL database with respect to a source Relational one.

1 INTRODUCTION

Due to the increasing amount of data generated by user interactions on the Web or by big data requirements, some companies are focusing on using non-relational databases, usually referred to as NoSQL systems, standing for 'Not only SQL' (Han et al., 2011). This term has been used to categorize databases characterized by horizontal scalability, less constrained structure or schema-less, and faster access compared to traditional relational databases (RDBMS) (McMurtry et al., 2013).

Experts comment that despite the rise of NoSQL databases during the past years, NoSQL is not necessarily a replacement for relational databases (McMurtry et al., 2013). Instead, NoSQL databases comply with big or social data demands or specific projects which strain Relational ones. Nevertheless, sometimes, data belonging to Relational databases need to be transferred to NoSQL ones in order to be used in specific projects. However, the data conversion process is not trivial, since it is necessary to have considerable knowledge about the data models at hand.

In this scenario of structural heterogeneity, we define our research problem as follows:

> Let *RDB* be a Relational database and $NSDB = \{NSDB_1, ..., NSDB_n\}$ a set of databases belonging to NoSQL models, where each $NSDB_i$ uses one of the following NoSQL approaches $A = \{$*Key-value, Column, Document, Graph*$\}$. We need to establish conceptual mappings between *RDB* elements and the different data structures underlying $NSDB_i$ in such a way that *RDB* can be converted to $NSDB_i$.

With this in mind, and considering the structural heterogeneity between Relational and NoSQL models, this paper presents the R2NoSQL approach for converting data between the referred models. To this end, it compares the data structures belonging to the Relational model with the four main NoSQL approaches (key-value, columns, documents and graphs), identifying a set of possible conceptual mappings between *RDB* and a $NSDB_i$. Also, it provides a tool prototype, which implements a case study with a Relational database and a Document based NoSQL system. Experiments have been done to evaluate the consistency of the generated mappings by analysing the results obtained from the same set of queries executed on both systems.

This paper is organized as follows: Section 2 introduces some concepts; Section 3 presents the approach; Section 4 describes some obtained results. Related work is discussed in Section 5. Section 6 draws our conclusions and points out future work.

2 NoSQL MODELS

NoSQL systems are a category of databases that do not follow principles of the Relational Model (Han et al., 2011). The term "NoSQL" does not relate to a

Freitas, M., Souza, D. and Salgado, A.
Conceptual Mappings to Convert Relational into NoSQL Databases.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 174-181
ISBN: 978-989-758-187-8

specific data model, but to a group of data models that differ from the relational approach and may have in common some features such as: they are usually open-source, distributed and horizontally scalable, and they present schema flexibility or even no schema (Han et al., 2011). NoSQL systems are classified in some categories in which the four main are: Key-value, Columns, Documents, and Graph. Indeed, their implementations may differ from each other, even when the systems belong to a similar category. In order to base our descriptions, we have chosen one example of each NoSQL category. They are briefly discussed in the following.

2.1 Key-value Model

The Key-value Model is the one with the simplest representation. Its structure consists of a list of pairs composed by a key and a value (Istvan et al., 2013).

Usually, Key-value NoSQL implemented systems allow, besides simple data types (e.g., numerals and strings), the use of lists and sets of values of simple types. This is what happens, for instance, in Redis (Redis, 2015), our example of Key-value system. A Key-value system such as Redis tends to support large volumes of data. Since it does not present data schemas, the developer may, by hand, introduce some metadata by naming the keys. On the other hand, it does not support queries to be performed on the data, but only on the search keys. Thus, all access is done through the search keys and only with the key it is possible to access the value. This access usually is accomplished with lower response times, one of its main benefits. This model does not support relationships in terms of reference keys and no referential integrity constraint.

2.2 Column Model

At a first sight, this model may be considered as similar to the Relational one, since it is also organized in terms of rows and columns. However, this approach deals with data in a non normalized way, i.e., by allowing nesting of tables inside tables (Lakshman and Malik, 2010). In this approach, rows do not store a tuple, but a set of attributes of the same type, while the set of attributes of a column contains the information from a given instance. Such feature allows queries to be performed more efficiently, although when recovering a complete instance it may become more costly.

Another important concept regards a “family of columns”, which means a set of instances of a given entity. In this structure, it is possible to have non-atomic attributes through the representation of value lists. Instances may have a different number of attributes, since there is no need to book storage space for null values. Also, there is no need to use join operations in order to query diverse entities.

2.3 Document Model

In Document Model, the data entities are grouped in documents as objects, which are composed by keys (properties) and values. These documents are usually serialized in JSON syntax (McMurtry et al., 2013).

A document is a collection of objects that are related to a data instance. The various documents belonging to the same data domain are stored in a collection of documents.Considering the MongoDB document system (Mongo, 2015), an instance key (called as an “objectId”) can be set at persistence time, or may have its value generated randomly by the database. It can provide uniqueness values for other fields by the specification of an index.

This model allows more complex queries involving different collections of documents. To this end, it is necessary that a document has a DBRef (Database Reference) to another related document or establish a reference. Despite allowing references, DBRef does not guarantee referential integrity constraint. A query may consider these references or use data embedded within the same document.

2.4 Graph Model

The Graph model is mainly concerned with representation and access, where data items are connected by relationships by means of a graph structure (McMurtry et al., 2013). The elements underlying a graph are, namely (McMurtry et al., 2013): nodes, edges and properties. Nodes correspond to data instances, edges refer to maintained relationships among node instances, and properties relate to data values. Some systems of such category allow the definition of their properties with the guarantee of unique values. One example regards the Neo4j system (Neo4j, 2015).

Nodes and edges can contain labels (terms which indicate a category) that classify them into more specific groups. For the nodes, these labels can be used to differentiate instances. On edges, labels may also be used to determine the type of relationship that is occurring.

An edge has an input node and an output node linking them. This feature, besides supporting references, also guarantees referential integrity by ensuring that the input node always makes reference

to the output node. The access keys to the nodes are automatically set by the system. However, it is possible to establish unique constraints for other node properties.

3 THE R2NoSQL APPROACH

In this section, we present some definitions along with the proposed approach.

3.1 Some Definitions

At first, we provide some definitions regarding the concepts underlying E-R and Relational models that we consider in our approach. Since we need to think about mappings between concepts, we define what we consider by "Concept" in each one of the working data models. Regarding the E-R Model, we may summarize a Concept as follows.

Definition 1 – E-R Concept. The set of concepts of an E-R conceptual model we are dealing with are C_E = {*Entity, Simple Attribute, Multi-valued Attribute, Composed Attribute, Relationship, Specialization*}.

In the light of the Relational Model, we define a Concept as follows.

Definition 2 – Relational Concept. Concepts of a relational structure are C_R = {*Table, Simple Attribute, Primary Key (PK), Foreign Key (FK)*}.

As discussed in Section 2, we observe that each NoSQL database model has specific data structures. Thereby, we provide the definition of the main Concepts of the four categories of NoSQL systems described previously.

Definition 3 – Key-value NoSQL Concept. A concept in a Key-value Model may be C_K = {*Search Key, Value, Value List, Value Set*}.

Definition 4 – Column NoSQL Concept. In a Column Model, a concept may be C_C = {*Column Family, Line, Column, Value Set, Value List, Primary Key (PK)*}.

Definition 5 – Document NoSQL Concept. In a Document Model, a concept may be C_D = {*Document Collection, Document, Field, Embedded Field, Field List, ObjectId, DBRef*}.

Definition 6 – Graph NoSQL Concept. A Graph model concept C_G = {*Label, Data Node, Property, Property Set, Id, Edge, Value Set, Value List*}.

Data indeed are instantiated differently in each one of the referred databases. Nevertheless, we can think about a data item or a data instance, in a general way, as follows.

Definition 7 - Data Item. A data item is an instance or an individual of a real entity in the data set at hand.

In the Relational Model, a data item is a tuple. In NoSQL approaches, it can be a data node, a data document, a column of a column family or simply a value from the key value model.

3.2 Our Proposal

As discussed in the previous sections, each database model has specific data structures and concepts, what provides structural heterogeneity conflicts among them. These conflicts occur because different choices of construct representation or integrity constraints are adopted in accordance with the options underlying each data model. Thereby, in this work, the task we are dealing with is concerned with what is necessary to convert concepts of a given *RDB* (a Relational database) to a $NSDB_i$ (a NoSQL one). Thus, it becomes necessary to specify conceptual mappings between concepts $C_r \in RDB$ and concepts $C_n \in NSDB_i$.

Our proposal, named as R2NoSQL approach, is based on three aspects: (i) defining conceptual mappings between *RDB* and $NSDB_i$; (ii) using these conceptual mappings to allow metadata and data conversion between the referred databases, and (iii) classifying source tables to help understanding their meaning in the database design. In the following, we provide the definitions underlying these issues.

3.2.1 Conceptual Mappings

Our approach deals with the structural heterogeneity of the data models and some aspects of database design. In order to cope with these issues, our mapping language handles the different existing concepts, which belong to the data models, but as a design reference, we consider some concepts not only from the Relational model but also from the E-R conceptual model. Thereby, we consider concepts from the Conceptual E-R model, which are not directly implemented in a Relational database, but they are close to real world and can be implemented in NoSQL systems. This conceptual mapping is the base for our conversion solution and without it, the process could not happen. These concepts regard particularly the composed and multi-valued attributes, and also specializations. By establishing that, we deal with a source model and a target model and we define the set of possible source Concepts, to be considered in a Mapping, as the following:

Definition 8 – Source Concept. A source concept C_s is the set of possible E-R or Relational concepts which may be used to compose a Mapping. Thus, $C_S = C_E \cup C_R$. Proceeding with the union operation, the final set results in C_S = {*Entity, Simple Attribute, Multi-valued Attribute, Composed Attribute, Relationships, Specialization, Table, Primary key (PK), Foreign key (FK)*}. Since a conceptual *Entity* always results in a relational *Table*, we abstract both ones only in the concept *Table*.

In the same way, we establish a target Concept, as the following.

Definition 9 – Target Concept. A target concept C_T is the set of possible NoSQL concepts which may be used to compose a Mapping. $C_T = C_K \mid C_C \mid C_D \mid C_G$.

Thus, the set of target concepts is composed by the possible concepts which belong to one of the NoSQL systems instantiated by a specific model.

With these definitions in mind, we define, firstly, in a general way, a Conceptual Mapping, as follows.

Definition 10 – Conceptual Mapping. A conceptual mapping *M* represents an association between a concept C_S and a concept C_T of a given $NSDS_i$, where $NSDS_i \in$ A, and A = {*Key-value, Column, Document, Graph*}. *M* defines a level of similarity between C_S and C_T.

A conceptual mapping *M* may be understood as a way of converting a given C_S into a C_T. Depending on the target $NSDS_i$, to a given C_S, there may be no corresponding C_T, i.e., there may be no concept in the target model that can be used for data conversion. When this fact happens, we point it as an empty or non existing target concept (∄).

Based on the previous definitions, we establish specific conceptual mappings between C_S and C_T, according to the $NSDS_i$ at hand. To this end, we consider the possibility of employing a table denormalization technique, which is the process of adding redundant data or grouping data, previously fragmented in a number of relational tables. Thereby, we may have nested tables or multi-valued attributes in one or more target structures, which may be sets, lists, documents or other ones, depending on the data model.

Let *RDB* be a source database composed by C_S and $NSDB_K$, a Key-value NoSQL system, composed by C_K. Specific structural conceptual mappings between C_S and C_K may be defined, as follows.

$RDB:Table \equiv NSDB_K: \nexists$
$RDB:SimpleAttribute \equiv NSDB_K:Value$
$RDB:ComposedAttribute \equiv NSDB_K:ValueList$
$RDB:Multi\text{-}valuedAttribute \equiv NSDB_K:ValueList$
$RDB:PK \equiv NSDB_K:SearchKey$
$RDB:FK \equiv NSDB_K: \nexists$
$RDB:Specialization \equiv NSDB_K:ValueSet$

Regarding data items, we may establish a mapping in the following way:

$RDB:DataItem \equiv NSDB_K:Value \mid NSDB_K:ValueList$

Although there is no corresponding concept to a *Table*, it is possible to simulate such concept by using composed search keys. In this case, keys are composed by a prefix together with the name of a given property, in such a way that we may identify to which entity it is associated. Indeed, it is not a defined standard, but one of our proposals.

The representation of relationships occurs with the storage of the search key values of a given data item inside another one. In many-to-many relationships, this happens in both sides of the data items at hand.

Now let *RDB* be a source database composed by C_S and $NSDB_C$, a Column NoSQL system, composed by C_C. Specific structural conceptual mappings between C_S and C_C are defined, as follows.

$RDB:Table \equiv NSDB_C:ColumnFamily$
$RDB:SimpleAttribute \equiv NSDB_C:Column$
$RDB:ComposedAttribute \equiv NSDB_C:ValueSet$
$RDB:Multi\text{-}valuedAttribute \equiv NSDB_C:ValueList$
$RDB:PK \equiv NSDB_C:PK$
$RDB:FK \equiv NSDB_C: \nexists$
$RDB:Specialization \equiv NSDB_K: ValueSet$

Regarding data items, we may establish a mapping in the following way:

$RDB:DataItem \equiv NSDB_C:Line$

In terms of relationships, a $NSDB_C$ allows their implementation by two options: (i) a column family may compose information from different but related tables; or (ii) data items may have a reference to other data items by storing the target primary key. The former is the most common option, since it allows a better response time.

Now let *RDB* be a source database composed by C_S and $NSDB_D$, a Document NoSQL system, composed by C_D. Structural conceptual mappings between C_S and C_D are defined, as follows.

$RDB:Table \equiv NSDB_D:DocumentCollection$
$RDB:SimpleAttribute \equiv NSDB_D:Field$
$RDB:ComposedAttribute \equiv NSDB_D:EmbeddedField$
$RDB:Multi\text{-}valuedAttribute \equiv NSDB_D:FieldList$
$RDB:PK \equiv NSDB_D:ObjectId$
$RDB:FK \equiv NSDB_D:DBRef \mid NSDB_D:EmbeddedField$
$RDB:Specialization \equiv NSDB_D: EmbeddedField$

Regarding data items, we may establish a mapping in the following way:

$RDB:DataItem \equiv NSDB_D:Document$

Relationships are implemented by defining object references between objects belonging to documents. Thereby, queries may take into account these references to get related objects information.

Now let *RDB* be a source database composed by C_S and $NSDB_G$, a Graph NoSQL system, composed by C_G. Specific structural conceptual mappings between C_S and C_G are defined, as follows.

RDB:Table ≡ $NSDB_G$*:LabelNode*
RDB:SimpleAttribute ≡ $NSDB_G$*:Property*
RDB:ComposedAttribute ≡ $NSDB_G$*:ValueSet*
RDB:Multi-valuedAttribute ≡ $NSDB_G$*:ValueList*
RDB:PK ≡ $NSDB_G$*:Id*
RDB:FK ≡ $NSDB_G$*:Edge*
RDB:Specialization ≡ $NSDB_K$*:ValueList*

Regarding data items, we may establish a mapping in the following way:

RDB:DataItem ≡ $NSDB_G$*:Node*

In the next section, we provide the way we classify identified tables in a given *RDB*.

3.2.2 Table Classification

Regarding the set of concepts $C_R \in RDB$, the main one is always a *Table*. Since a table may be the result of E-R conceptual entities, relationships, specializations, multi-valued or composed attributes, we need to understand what a *Table* means, and its importance, to the *RDB* at hand. We have defined a classification of the source Tables as follows.

- **Main Tables:** These are the main tables in a RDB design. They usually correspond to entities found in the conceptual model.
- **Subclasses:** These tables are a complement to the definition of a main table. They represent specializations of the main tables, but do not exist independently.
- **Relationships:** This classification typifies a specific kind of table, which implements a many-to-many (N:N) relationship in the conceptual model.
- **Common Tables:** The other types of tables are defined as common in a source RDB schema.

Our proposed algorithms take into account such table classification in order to identify the conceptual mappings to be used.

3.2.3 Conversion Algorithm

Based on the specified conceptual mappings between C_S and C_T, and on the table classification, some algorithms have been developed to allow data conversion. A main algorithm, named as *Algorithm1-Data Conversion*, receives a *RDB* enriched with a Table Classification as input and generates a $NSDB_i$ as output.

Algorithm1: Data Conversion.

```
Input: RDB rel;
Output: NSDSi ns;
Begin
    //Looks for main tables
1:  For Each table of rel Do
2:  If (table.classification() is "main")
    Then
3:      get all table attributes to object
4:      For Each table referencing table Do
          //looks for related tables
5:        goDeep(table, object);
            //persists object on ns
6:        persist(object) ;
7:      End For;
8:    End If;
9:  End For;
End Data_Conversion;
```

In our approach, the input RDB is composed by its metadata and data. Tables were already classified and this classification is also used as input. Based on that, the algorithm verifies the kinds of existing tables and, for each type, extracts the set of data items. Through references (FKs), data tables are traversed and analyzed. At such verification time, decisions are taken according to each type of table.

Algorithm2: goDeep.

```
Input: table, object;
Output: table, object;
Begin
    //Looks for tables that reference table
1:  table2 := findTableDeep(table);
    2:  Do Switch table2.classification
3:    case "common":
4:      get all table2 attributes to object;
5:      goDeep(table2);
6:    case "subclass":
7:      get all table2 attributes to object;
8:      goDeep(table2);
11: End Switch;
    //looks for related tables
10: goUp(table, object);
End goDeep;
```

Main tables constitute the basis for the analysis. With a main table at hand, the algorithm verifies if it is related with other ones. In this case, it calls another algorithm (Algorithm2–goDeep) where existing relating tables are identified. Regarding these selected tables, some options may be taken, namely: (i) If the table is another main table, no other procedure is accomplished because, later, the opposite direction of the relationship will be considered. At this later time, the second table will refer to the first one; (ii) If the table is a relationship

table, no action is taken because, only when all the data items are persisted, relationships can be analyzed; (iii) If the table is a common one, it is possible to add its attributes in that main table as part of its structure; (iv) If the table is a subclass, then its attributes are added to the main table. It is understood that it is a specialization of the main one.

Each time a new table is found in-depth analysis, the algorithm selects this table and repeats the process until there are no more tables, or a stop condition happens. This process is responsible for the denormalization of the data, in which the data related to the main table are included in a data item.

Algorithm3: goUp.

```
Input: table, object;
Output: object, reference_list,
referenced_list;
Begin
    //Looks for tables that reference table
1:  table2 := findTableUp(table);
2:  Do Switch table2.classification
3:    case "common":
4:      get all table2 attributes to object;
5:      goUp(table2);
6:    case "subclass":
   //saves data items to set relationship
7:      reference_list.add(table);
8:      referenced_list.add(table2);
8:      goDeep(table2);
9:    case "relationship": break;
10:   case "main":
11:     reference_list.add(table);
12:     referenced_list.add(table2);
13:  End Switch;
End goUp;
```

After the identification of in-depth relationships, the tables that the main table refers are searched up (*Algorithm3 – goUp*). According to the identified relationships, one of the following options will be considered: (i) to capture the attributes and move up or (ii) to save identified instances in a list. This function separates tables in which there could be composed or multi-valued attributes. It may also show that a new entity has been found, and a relationship should happen. The procedure is repeated until a stop condition is reached.

After the main table and its related tables of a data item are analyzed, the whole set of attributes is persisted as an entity in the target database (*Algorithm1*, line 6). Just after all instances have been persisted, the algorithm must define the relationships among them.

To establish the one-to-one or one-to-many relationships, generated lists (when a main table mentioned another one) are used. These lists are included in the data items that have references, and the data items that are referenced. For each type of implemented target database, the algorithm must implement a specific procedure. In this work, we show one regarding the MongoDB system. This algorithm, named as *oneTo,* was implemented to provide DBRef storage. It stores in the corresponding document of Table 1 a reference to list2, and in the one corresponding to Table 2, a reference to list1. In terms of many-to-many relationships, it is necessary to identify the double meaning of these references (*manyToMany* algorithm). In MongoDB case, a DBRef of each document involved in the corresponding document is stored. For instance, the student Bill Gates held a publication. Thus there is a publication document reference in the student's document, and there is a student document reference in the publication document. Other solution would be to embed all related data into one document. However, this approach can let the execution of queries harder. For instance, if student documents contain publication information, more effort would be necessary to retrieve people involved in a specific publication.

4 RESULTS AND EXPERIMENTS

In this section, we present some implementation and experimental results.

4.1 Implementation and Example of use

We have developed the R2NoSQL approach in the Java language. The main functional requirements underlying the tool's development are the following:

- **Set Source Database:** it includes the definition of the relational DBMS to be used as well as the metadata and data extraction step.
- **Classify Table:** The tables extracted from the Relational database will be classified by the user.
- **Set Target Database:** The user chooses the target NoSQL database.
- **Execute Data Conversion:** It analyzes the extracted metadata and data from the source database, verifies tables' classification and possible conceptual mappings, identifies the target NoSQL concepts and persists corresponding data in the target database.

In this current version, we have developed a prototype which deals with a *RDB* (e.g., MySQL) as a source database and a $NSDB_D$ as a target one. To the latter, we have used the MongoDB system.

We provide an example in the following. As source *RDB*, we have used a database with 15 tables

(e.g., Person, Publication, Student). Among them, there are some relationships (e.g., between Person and Publication), and some specializations (e.g., Student is a specialization of Person).

The Table classification is accomplished by the user, since, in this version, we have a semi-automated tool with respect to this functionality. Thus, after extracting the source *RDB* metadata, the tool asks the user to classify the extracted tables.

After table classification, the R2NoSQL tool is able to proceed with the data conversion, in accordance with the defined algorithms (Section 3.2.3). The tool selects instances of each table classified as main, storing the table attributes and their associated values. This happens according to the existing mappings. When this process finishes, the tool looks for instances related to what was analyzed as a complement to the main tables. These related tables can be a representation of a complex value as a composed attribute, a multi-valued attribute or even a specialization.

Considering a fragment of the source database at hand, we show some instances belonging to tables Person and Student in Table 1 and Table 2, respectively. Taking into account the data presented in Table 1 and Table 2, the tool produces a document as depicted in Figure 1. The resulting collection of documents is named by using the name of the main table of that entity. The attributes and values of the tables were converted into document fields. However, new fields were introduced as depicted in Figure 1: AR_master regards the key attribute of table Master, a specialization of Student; Person_X_Publication represents the many-to-many relationship between Person and Publication.

4.2 Experiments

We have conducted some experiments to verify the effectiveness of our approach. The goal of our experiments was twofold: (i) to check whether a set of queries formulated and executed in a *RDB* may be either formulated and executed in a generated $NSDB_D$, and (ii) to verify if the query results obtained from both databases are similar, i.e., if the used mappings and algorithms have generated a consistent target $NSDB_D$. In order to verify the latter goal, we checked both obtained query results and compared the ones obtained in a $NSDB_D$ with respect to the corresponding ones from a RDB.


```
{
    "_id": {
        "$oid": "55a6a2cc13044b9cd70b4f43"
    },
    "CPF": "74852963214",
    "RG": "10987312",
    "name": "Bill Gates",
    "Bdate": "10-28-1955",
    "Natural From": "Washington",
    "nationality": "American",
    "e_mail": "gates@ms.com",
    "url": "http://www.gatesnotes.com/",
    "user": "gates",
    "pword": "gates",
    "profile": "U",
    "type": "master",
    "AR": "790099",
    "Additional_info": "Co-author of 19 articles.",
    "AR_master": "790099",
    "Adm_semester": "1",
    "Adm_year": "1981",
    "Egressiondate": "03-19-1981",
    "Person_X_Publication": [
        {
            "Publication": {
                "$ref": "Publication",
                "$id": {
                    "$oid": "55a6a2d413044b9cd70b4f4c"
                }
            }
        }
    ]
}
```


Figure 1: Document obtained after data conversion.

The working data scenario was the same presented in the example of Section 4.1. This database was populated with 45 tuples, and queries were specified to be executed over them. Table 3 shows an examples of query used in our experiments, which shows all professor attributes according to his name. It is presented in SQL and in MongoDB query language.

The first experiment goal was accomplished, since, by using the tool, we could submit and execute the same set of queries in both source and target databases.

Regarding the latter goal, we have compared the obtained query results in terms of data items (instances) and their properties. For each query submitted and executed in MongoDB, we measured the degree of similarity of the answers with respect to the set of answers obtained in the relational database (which acted as a gold standard).

All queries returned similar instances with identical property values (100%). Differences were

Table 1: Table *Person* with a tuple.

CPF	RG	name	Bdate	Natural From	nationality	e_mail	url	user	pword	profile
74852963214	10987312	Bill Gates	10-28-1955	Washington	American	gates@ms.com	http://gatesnotes.com	gates	gates	U

Table 2: Table Student with a tuple.

AR	CPF	Additional_info	Adm_semester	Adm_year	Egressiondate
790099	74852963214	Co author of 19 articles.	1	1981	03-19-1981

obtained only in terms of query results presentation, but not regarding the set of obtained data items.

Therefore, we can see that the goals of this experiment were achieved. It was possible to require the same information from both source and target databases. In addition, it was possible to obtain the same set of query results from both ones.

Table 3: A query example used in the experiment.

SQL	MongoDB query language
select pe.*, pf.* from Person pe inner join Professor pf on pe.cpf=pf.cpf inner join IC_Professor i on pf.cpf=i.cpf where pe.cpf = '95175368429' union select pe.*, pf.* from Person pe inner join Professor pf on pe.cpf=pf.cpf inner join Invited_Professor ip on pf.cpf= ip.cpf where pe.cpf = '95175368429'	db.Person.find({cpf: "98632541754", $or:[{type: "ic"},{ type: "invited"}]}, {cpf:1, rg:1, name:1, birth_date:1, naturalness:1, nationality:1, user:1, password:1, profile:1, e_mail:1, type:1, additional_info:1})

5 RELATED WORK

Data conversion approaches regarding Relational and NoSQL models have been tackled. Zhao et al. (2014) propose an automatic approach for converting relational database schemas to NoSQL ones, which establishes conceptual rules for the denormalization of the original data. Potey et al. (2015) provide a tool to perform data conversion, in which the target database is an equivalent relational schema in a Document structure. Karnitis and Arnicans (2015) instead provide a semi-automatic approach, which allows a comprehension of the relationships that the tables carry one over the other by a classification strategy. Mpinda et al. (2015) present a data conversion process that aggregates data tables, which are analyzed along with the established relationships.

Our proposal extends some of these concepts. We provide a denormalization technique and we deal with some kinds of conceptual relationships, by producing references when possible. We have a table classification strategy to enrich the overall process. Finally, our approach may be applied to any of the target NoSQL models.

6 CONCLUSIONS

We presented the R2NoSQL approach, which allows data conversion between relational and NoSQL databases. This approach is based on conceptual mappings defined between structural concepts from relational and NoSQL ones.

Experiments have shown that obtained NoSQL database is consisted with the source relational one, by executing the same set of queries in both source and target databases. In fact, they produced similar query results.

As future work, some enhancements will be done: (i) the tool will be extended to accomplish data conversion by considering other categories of NoSQL systems, and (ii) an automated query conversion process will also be taken into account.

REFERENCES

Han, J., Haihong, E., Le, G., Du, J., 2011. Survey on NoSQL database. In: *Pervasive computing and applications (ICPCA)*, 2011 6th international conference on. IEEE. p. 363-366.

Istvan, Z., Alonso, G., Blott, M., Vissers, K., A flexible hash table design for 10Gbps key-value stores on FPGAs. In: *Field Programmable Logic and Applications (FPL)*, 2013 23rd International Conference on. IEEE. p. 1-8.

Karnitis, G. and Arnicans, G., 2015. Migration of relational database to document-oriented database: Structure denormalization and data transformation. *Communication Systems and Networks (CICSyN)*, 7th International Conference on Computational Intelligence. p. 114–118.

Lakshman, A., Malik, P., 2010. Cassandra: a decentralized structured storage system. *ACM SIGOPS Operating Systems Review*, v. 44, n. 2, p. 35-40.

McMurtry, D., Oakley, A., Sharp, J., Subramanian, M., and Zhang, H., 2013. *Data access for highly-scalable solutions: Using sql, nosql, and polyglot persistence. Microsoft patterns & practices*.

MongoDB, 2015. Available at https://www.mongodb.org/. Last access on December, 2015.

Mpinda, S. A. T., Maschietto, L. G., and Bungama, P. A., 2015. From relational database to columnoriented nosql database: Migration process. *International Journal of Engineering Research & Technology (IJERT)*, 4. p. 399–403.

Neo4j, 2015. Available at http://neo4j.com. Last access on December, 2015.

Potey, M., Digrase, M., Deshmukh, G., and Nerkar, M., 2015. Database migration from structured database to non-structured database. *International Conference on Recent Trends & Advancements in Engineering Technology (ICRTAET 2015)*, p. 1–3.

Redis, 2015. Available at http://redis.io/. Last access on December, 2015.

Zhao, G., Lin, Q., Li, L., Li, Z. 2014. Schema Conversion Model of SQL Database to NoSQL. In: P2P, Parallel, Grid, *Cloud and Internet Computing (3PGCIC)*, 2014 Ninth International Conference on. IEEE. p. 355-362.

Identification of Organization Name Variants in Large Databases using Rule-based Scoring and Clustering
With a Case Study on the Web of Science Database

Emiel Caron[1] and Hennie Daniels[1,2]
[1]*Erasmus Research Institute of Management, Erasmus University Rotterdam, P.O. Box 1738, Rotterdam, The Netherlands*
[2]*Center for Economic Research, Tilburg University, P.O. Box 90153, Tilburg, The Netherlands*
caron@ese.eur.nl

Keywords: Large Scale Databases, Data Warehousing, Database Integration, Data Cleaning, Data Mining, Clustering.

Abstract: This research describes a general method to automatically clean organizational and business names variants within large databases, such as: patent databases, bibliographic databases, databases in business information systems, or any other database containing organisational name variants. The method clusters name variants of organizations based on similarities of their associated meta-data, like, for example, postal code and email domain data. The method is divided into a rule-based scoring system and a clustering system. The method is tested on the cleaning of research organisations in the Web of Science database for the purpose of bibliometric analysis and scientific performance evaluation. The results of the clustering are evaluated with metrics such as precision and recall analysis on a verified data set. The evaluation shows that our method performs well and is conservative, it values precision over recall, with on average 95% precision and 80% recall for clusters.

1 INTRODUCTION

In many databases, one organisational entity is listed in the records with many associated name variants. For example, the Leiden University (2015) has many name variants in the Web of Science (WoS) database, like: University Leiden, Leiden Universiteit, Leiden State University, State University Leiden, Leiden University Hospital, State University Leiden Hospital, Leiden Universitair Medisch Centrum, LUMC, and so on. Large companies often have many name variants, e.g. the technology company Royal Philips has several hundreds of name variants in the Patstat (2015) database, which is a statistical database filled with patent information. Obviously, manual normalisation of organisation names is not feasible in large databases, which might list millions of companies and organisations.

The research problem that is addressed here is: "How can organization name variants be identified automatically in large databases?". The answer to this problem is given by a general method for the identification of organization name variants using rule-based scoring and clustering proposed in this paper. The method is able to cluster name variants in large databases with millions of records in an efficient way. The emphasis of the method is on the cleaning of names not on unification. The results of this method are useful for any analysis involving correct and unified organisation names, such as: company patent analysis, evaluative bibliometrics and the ranking of scientific institutes, the assessment of cooperation and communication between organizations, and the creation of linkages, based on company names, between Customer Relationship Management databases.

Data cleaning is often the necessary step prior to knowledge discovery and business analytics. Automatic data cleaning methods can be categorised in several groups (Maletic and Marcus, 2010): transformational rules, statistical methods for numeric data, and data mining methods, such as cluster and pattern matching techniques (Cohen et al., 2003, Koudas et al., 2004, Morillo et al., 2013), for categorical data. Data mining methods for the identification of organisation name variants and person name disambiguation are divided into supervised and unsupervised learning approaches. In supervised learning approaches, a classifier is trained on a data set with pairs of records, where organisation with similar names are classified as being the same entity or a different entity. The problem with

Caron, E. and Daniels, H.
Identification of Organization Name Variants in Large Databases using Rule-based Scoring and Clustering - With a Case Study on the Web of Science Database.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 182-187
ISBN: 978-989-758-187-8

supervised approaches in this context, is that a large, manually checked, representative, data set is required for training. Such a data set is usually not available, which makes supervised approaches for our problem hard to use. In unsupervised learning, a metric of similarity is defined between pairs of records, that describe an organisational entity, and after that a clustering algorithm is applied (Levin et al., 2012, Song et al., 2007). The method described in this paper is based on unsupervised rule-based clustering, in combination with approximate string pattern matching. A clear advantage of our method is that the matching rules are easy to understand and combine.

The organization of this paper is as follows. In the next section, the phases of the method for the clustering of organisation names are explained in detail. After that the method is evaluated, with precision-recall analysis, on the WoS database for the clustering of scientific organisation names. We close the paper with some concluding remarks and proposals for further research.

2 METHODOLOGY

A visual summary of the process for the identification of organisation name variants is provided by Figure 1 in the Appendix. The method is composed out of three stages:

1. Pre-processing;
2. Rule-based scoring and clustering;
3. Post-processing.

Organisational meta-data from a source database is taken as input in the process and clusters of organisation name variants are produced as output. Typical examples of important meta-data available for organisation name matching are: country, city, postal code, street, organisation type, email domain, etc. The method is designed to cluster all organisation name variants in the whole database. In the case study the method is applied on the WoS database (version April 2013) with roughly 124 million publication records. Moreover, the method is implemented with a combination of Microsoft SQL Server and Visual Studio, where SQL server is used for the data handling and Visual Studio for the implementation of the cluster algorithm.

2.1 Pre-processing

In the pre-processing stage the relevant meta-data items are cleaned and harmonized to improve the data quality, and helper tables are created for the subsequent clustering stage.

Postal code data is cleaned an put into a consistent format. Besides, postal codes are classified into groups, indicating the number of different organisations present in a postal code area. Groups with a relative high number of organisations are treated under a stricter regime, e.g. with a higher threshold.

From the available data, the organisational types are determined, such as ‘company’, ‘bank’, ‘university’, ‘hospital’, ‘institute’, etc., with string extraction patterns and regular expressions.

An important data element for clustering, when it is available, is the email domain address that is linked with an organisation, because it is very discriminative. Usually, multiple email domains are connect to large organisations, e.g. Leiden University uses ‘leidenuniv.nl’, but also ‘liacs.nl’ and ‘lumc.nl’. In the pre-processing stage, the email domains are replaced by their most popular or recent variant. Email domains that cannot be directly linked to an organisation, e.g. ‘gmail.com’ or ‘hotmail.com’, are removed, because they cannot be used in an meaningful way.

2.2 Rule-based Scoring and Clustering

In this stage, the clusters are created that identify likely name variations of the same organization, in the following steps (see Figure 1):

a. A set of rules is created that produces pairs of organizational names with common characteristics and string name similarity. These rules target specific elements of the organizations’ characteristics such as combinations of the country, city, postal code, email address, organisation type.
b. A scoring system is applied on the rules and scores are computed for created record pairs.
c. Pairs above the threshold are clustered with an algorithm, taking linear time, that searches for connected components.

2.2.1 Apply Rules (Step 2a)

In step 2a, the objective is to create pairs of organisation name variants by self-joining the tables that result from the pre-processing stage. For this purpose scoring rules are used that are depicted in Table 1, where rules 1-4 are the basic rules and rules 5-9 are combinations of the basic elements. The score for a rule increases when it contains more meta-data elements, indicating that there is more proof that a record pair is a name variant of each other. Therefore

rule 9 is considered a stronger rule than rule 1. The number of rules and scores can easily be adapted if more relevant meta-data is available. The rule score values and the threshold values are based on domain expert knowledge about the database under consideration. The values are fine-tuned in the initial evaluation of the method on a verified data set.

The scoring system assigns scores to record pairs from the strongest to the weakest rules. If a record pair is scored on a strong rule, the pair is not considered for the weaker rules, to prevent additional scoring. Not listed in Table 1, is that there are additional constraints for each rule, like the size of the city, the type of postal code (general or specific), and so on, that are configured very strict for strong rules but are given more degrees of freedom for weak rules.

Table 1: Example rules with associated meta-data, organisation name similarity, and rule scores. The threshold value is 4 in the example.

Rules	Country	City	Postal code	Email	Org. type	Name sim.	Score
Rule 1	✓	✓	✓				1
Rule 2	✓	✓		✓			1
Rule 3	✓	✓			✓		1
Rule 4	✓	✓				✓	2
Rule 5	✓	✓	✓	✓			2
Rule 6	✓	✓		✓	✓		2
Rule 7	✓	✓	✓		✓		2
Rule 8	✓	✓	✓	✓	✓		4
Rule 9	✓	✓	✓	✓	✓	✓	10

Notice that rules always only hold in a specific country and city, because organisation names might not be unique in the whole world or even in a specific country. The rules use the meta-date: postal code information (rule 1), email domain (rule 2), organisation type (rule 3), and organisation name similarity (rule 4). For example, rule 1 matches organisation names records in a specific postal code area in The Netherlands for the city of Amsterdam.

Rule 1 specifies record pairs with postal codes that match exactly within a country and city. In this rule, the number of records an organizational name variant has in correlation with a specific postal code, is taken into account. This measure is important because an organization with only a few records assigned to a specific postal code area is suspected to be a false positive result. Another measure in this rule, is the percentage of records an organization name label has in a specific postal code, in relation to the total number of records associated with a certain organisation name. This percentage is also used to filter out organization name variants with low values on this measure.

Rule 2 is defined as record pairs with email domains that match exactly. The email domain labels should have a minimum count in the database. Records coupled by rule 3 share the same organisation type in a city. Organisation names that could not be typed in the pre-cleaning are excluded.

Rule 4 scores the level of string similarity of two organisation names within a city with the Levenshtein distance. The intuitive definition of Levenshtein distance is the amount of edits one needs to perform to change one organisation name into another organisation name. The implementation of Levenshtein distance described here uses the edit distance to calculate (in %) how similar is one string to another string. The use of the Levenshtein distance is based on the premise that organisational names that score above a certain threshold value, say 95% or so, can be considered as similar, and are therefore paired.

Rules 5-9 are combinations of rules 1-4. The combined rules have stricter thresholds than the basic rules, so the rules could match on different records.

2.2.2 Score Record Pairs (Step 2b)

Pairs of records are scored in step 2b. A record pair is described as two records that have scored on at least one rule and therefore share meta-data. Records can score on multiple rules, i.e. the scoring is additive. Therefore, the total score for record pairs has to be determined.

An example with 5 records, their active rules, and their total scores is presented in Figure 2. A line between two circles indicates a record pair. For example, the two records for ‘Vrije Univ Amsterdam’ and ‘VU Amsterdam’ share the same postal code, email domain, and organisation type, within the same country and city. Rule 4 does not fire, the string similarity is considered too low. Therefore, this record pair receives 4 points (see Table 1). The other records in the example are scored in the same way and are represented with a connecting line.

In step 2c, record pairs above the threshold value, total scores ≥ 4 in Figure 2, are included for the clustering algorithm. The threshold value is increased for geographical areas with a high number of organisations, to prevent the potential erroneous coupling of records pairs. The rules scores express the strength of a certain rule. Furthermore, the more rules that are active for a publication pair, the more evidence there is that two different organisation names are indeed variants of each other. In the example scoring system, only rules 8 and 9 are strong enough to solely pass the threshold value. However, for a pair of records often combinations of rules are required to exceed the threshold value., e.g. see the link between ‘Univ Amsterdam’ and ‘Univ Hosp Amsterdam’ in Figure 2.

2.2.3 Cluster Records (Step 2c)

Matched records pairs, i.e. record pairs with a score above the threshold, are clustered by means of single-linkage, hierarchical, clustering in step 2c. In Figure 2, for example, the records 'Univ Amsterdam' and 'Univ Hosp Amsterdam' are a matched pair, and the records 'Univ Amsterdam' and 'Emma Childrens Hosp' are a matched pair. The clustering algorithm makes a link between these two initial clusters via the joint record 'Univ Amsterdam', by merging the two clusters into a new cluster with three records, depicted by 'Cluster 2' in Figure 2, and so on. The final cluster will represent the (partial) history of name variants of an organisation. In the figure, there is not enough proof for the clustering of 'VU Amsterdam' with 'Univ Hosp Amsterdam', this is indicated by a dotted line. Therefore, two clusters are created by the algorithm, representing the two different universities in the city of Amsterdam. Notice that, if the threshold is increased, e.g. more clustering is induced, resulting in on average smaller cluster sizes.

2.3 Post-processing

In the post-processing stage, non-clustered records are labelled as separate clusters and added to the results to give a complete overview. Finally, tables are created that provide detailed summary information about the clusters. An example of such a table, which gives a cluster description, is given in Table 2. A combination of relational support tables, provide a good basis to work with the results of the clustering in practical data analysis.

3 CASE STUDY

In this case study, the method is used for the cleaning of scientific organizations present in the Web of Science (2015) bibliographic database. Bibliometric databases are large databases that are used to study the growth of scientific publications, patterns of collaboration, the impacts of science, and evidence-based performance assessment. For most of these analyses, it is necessary to increase the data quality by cleaning the relevant tables.

Cleaned organizational names are important for the Leiden Ranking (2015), produced by the Centre for Science and Technology Studies (CWTS, 2015). The CWTS Leiden Ranking 2015 offers insights into the scientific performance of 750 major universities worldwide, based on indexed research publications obtained from the Web of Science. This university name identification process is carried out manually and is therefore time-consuming and cumbersome. In the manual process, organizational labels are clustered and after that unified. This method can be trusted as very accurate because every organizational label that is under investigation, is verified with the help of the Internet and with other means, in order to be concluded as a name affiliation of a certain scientific organization. These cluster are a 'golden set', and used as a benchmark for the clusters produced by the automatic method in a precision-recall analysis.

In Table 2, the partial cluster for Leiden University in The Netherlands is depicted to show the end product of the clustering method. Each cluster is identified by a 'cluster_id' and is composed out of one or more records, that show supportive meta-data.

Table 2: Example cluster with id '3717' with name variants for 'Leiden University', ordered by the number of scientific publications, labelled by 'n_pubs'.

cluster_id	nu	ny	nc	nc_no	n_pubs	org_type
3717	NETHERLANDS	LEIDEN	LEIDEN UNIV	6739	69006	Univ
3717	NETHERLANDS	LEIDEN	LEIDEN STATE UNIV	3753	3701	Univ
3717	NETHERLANDS	LEIDEN	LEIDEN UNIV HOSP	20484	3480	Univ
3717	NETHERLANDS	LEIDEN	STATE UNIV LEIDEN	853	2919	Univ
3717	NETHERLANDS	LEIDEN	UNIV LEIDEN HOSP	42225	2231	Univ
3717	NETHERLANDS	LEIDEN	LUMC	550897	1505	Univ
3717	NETHERLANDS	LEIDEN	UNIV HOSP LEIDEN	7657	1251	Univ
3717	NETHERLANDS	LEIDEN	Leiden Univ Med Ctr	780691	1096	Univ
3717	NETHERLANDS	LEIDEN	UNIV MED CTR	177905	189	Univ
3717	NETHERLANDS	LEIDEN	UNIV MED CTR LEIDEN	216329	115	Univ
3717	NETHERLANDS	LEIDEN	Leids Univ	764733	95	Univ
3717	NETHERLANDS	LEIDEN	Leids Univ Med Ctr	1045872	49	Univ
3717	NETHERLANDS	LEIDEN	Leiden Univ Med Ctr LUMC	1740207	26	Univ

The precision and recall performance values for the best clusters per scientific organisation in the golden set are depicted in Figure 3, where the organisation names on the x-axis are ranked based on precision-recall values. The cluster with the highest value for the F1 measure, defined as the harmonic mean of precision and recall, is taken as the best cluster. In addition, the numbers in Table 3 show on average a precision of 0.95 and a recall of 0.80 for the best cluster.

Table 3: Average values of evaluation metrics for the best cluster in the Leiden ranking data set.

	Precision	Recall	F1
Best cluster (mean)	0.95	0.80	0.84
Best cluster (median)	1.00	0.89	0.98
Best 3 clusters (mean)	0.91	0.86	0.83

This shows that the clustering method is conservative, it chooses precision above recall. If the 3 best clusters for an organization are used in the evaluation the average recall is pushed to 0.86, with a slightly lower average precision. This indicated that for a number of

organisations the name variants are spread over a number of accurate clusters. Clusters with a lower precision are, in general, clusters belonging to very large cities, where multiple research institutes can be found in a relatively small area, which makes name normalisation more difficult.

4 CONCLUSIONS

In this research we have presented an efficient general rule-based scoring method for the clustering of name variants of organizations in large databases. The rules are based on organisation name similarity and meta data in the context of the organisation, like: country, postal code, email domains, organization type, etc. Basically, the method can work with any piece of relevant meta-data, as long as it is shared between records. Multiple rules can be combined to link organization names, because of the scoring system. The more rules that hold for a pair of organisation names, the more evidence there is that the organisation names are indeed valid name variants of each other. In other words, the rules in the system strengthen each other. Moreover, the rules are easy to understand and combine. Incorrect matching of organisation names is partly prevented by lowering the scores for certain sensitive rules and by increasing the threshold values, for example, for geographic locations with a high number of organisations.

Based on the results of the case study, it can be stated that the clustering method is careful, it values precision (on average 95%) over recall (on average 80%). In general, precision and recall are lower for areas with a high number of scientific organisations. Name variants of organizations might be split over multiple clusters, if there is not enough evidence for coupling names variants together. However, these alternative clusters do have a high precision and are therefore useful for analysis.

In conclusion, the method can be viewed as a general method for data cleaning, because it can be used to other types of data, e.g. person or author name disambiguation (Caron and Van Eck, 2014), as long as there is relevant meta data available. In future research, the cleaning method should be tested on multiple databases with name variants to find optimal values for scores and thresholds, and to improve the quality of the method for very large cities. In addition, we want to push recall performance forwards by further integrating string similarity measures (Cohen et al., 2003) in the method.

ACKNOWLEDGEMENTS

I thank Vasileios Stathias and Nees Jan van Eck for their contributions to this research. In this study I used the database facilities of the Centre for Science and Technology Studies (CWTS, 2015).

REFERENCES

Caron, E., van Eck, N.J., 2014. Large scale author name disambiguation using rules-based scoring and clustering, *In Proceedings of the 19th International Conference on Science and Technology Indicators*, pages 79-86, Leiden, The Netherlands.

Cohen, W., Ravikumar, P., & Fienberg, S., 2003. A comparison of string metrics for matching names and records. In *KDD Workshop on Data Cleaning and Object Consolidation*, Vol. 3, pp. 73-78.

CWTS, 2015. *Centre for Science and Technology Studies*, http://www.cwts.nl, Leiden, The Netherlands.

De Bruin, R., Moed, H., 1990. The unification of addresses in scientific publications. *Informetrics*, 89/90, 65-78.

Koudas, Nick & Marathe, A. & Srivastava, D., 2004. Flexible string matching against large databases in practice. *Proceedings of the 30th VLDB Conference*.

Leiden Ranking, 2015. *CWTS Leiden Ranking 2015*, http://www.leidenranking.nl, The Netherlands.

Leiden University, 2015. http://www.leidenuniv.nl, Leiden, The Netherlands.

Levin, M., Krawczyk, S., Bethard, S., & Jurafsky, D., 2012. Citation-based bootstrapping for large-scale author disambiguation. *Journal of the Association for Information Science and Technology*, 63(5), 1030-1047.

Maletic, J. I., & Marcus, A., 2010. Data cleansing: A prelude to knowledge discovery. In *Data Mining and Knowledge Discovery Handbook* (pp. 19-36). Springer.

Morillo, F., Santabárbara, I., & Aparicio, J., 2013. The automatic normalisation challenge: detailed addresses identification. *Scientometric*s, 95(3), 953-966.

Patstat, 2015, EPO Worldwide Patent Statistical Database, http://www.epo.org.

Song Y., Huang J., Councill I., Li J., & Giles C., 2007. Efficient topic-based unsupervised name disambiguation. In *Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries* (JCDL '07). ACM, New York, NY, USA, 342-351.

Web of Science, 2015. *Thomson Reuters*, United States. http://www.webofscience.com.

APPENDIX

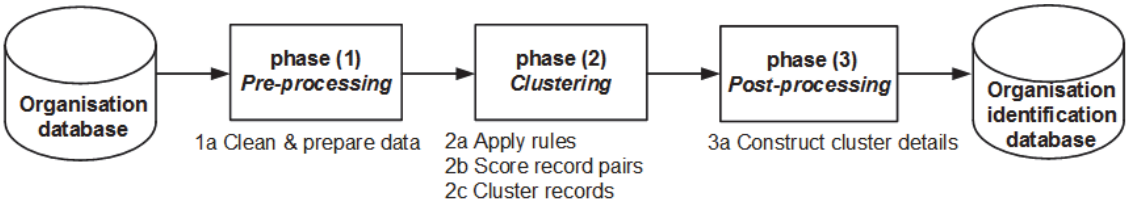


Figure 1: Stages in the identification process of organization name variants.

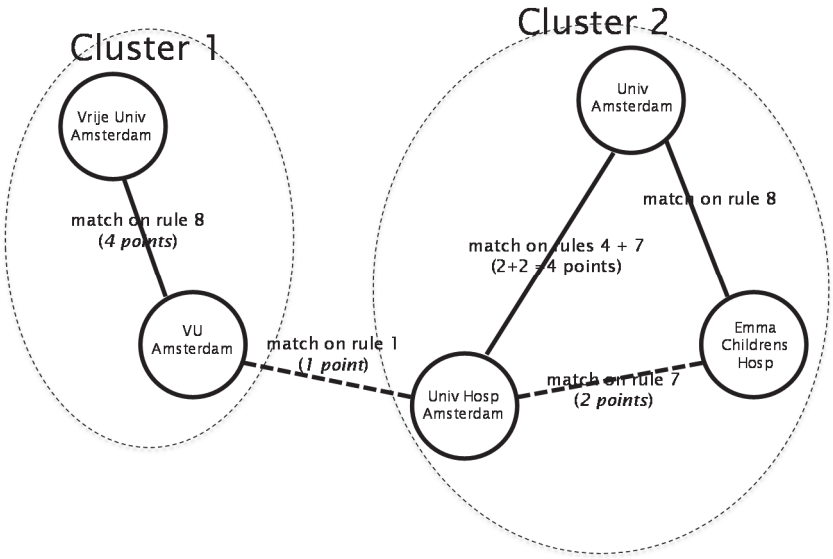


Figure 2: Scoring and clustering example for the city of Amsterdam (with threshold ≥ 4). Amsterdam has two universities the Vrije University Amsterdam (Cluster 1) and the University of Amsterdam (Cluster 2).

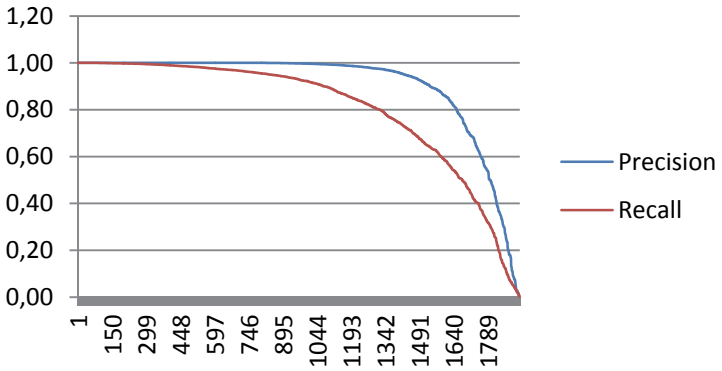


Figure 3: Precision (upper line) and recall (lower line) analysis on the Leiden Ranking data set.

The Impact of the Implementation of ERP Satisfaction of End Users in Major Moroccan Companies

Fatima Jalil[1], Abdellah Zaouia[1] and Rachid El Bouanani[2]
[1]*INPT: National Institute for Posts and Telecommunications, Rabat, Morocco*
[2]*Faculty of Law, Economics and Socials, Mohammedia, Morrocco*
{jalilfatima12, elbouanani8374}@gmail.com , zaouia@inpt.ac.ma

Keywords: Enterprise Resource Planning (ERP), User Satisfaction, Quality change, Information Technology, Information Systems.

Abstract: In recent years, the implementation of ERP is as a lever for development and inter-organizational collaboration. Despite the benefits of ERP, the success of their implementation is not always assured. The introduction of an ERP in a company requires organizational changes that may provoke resistance to cause adverse effects on the success of these projects. This article proposes a model and tests to evaluate the success of a system "Enterprise Resource Planning "(ERP) based on a measure of user satisfaction. Referring to the model DeLone & McLean and the work of Seddon & Kiew The criteria that can influence user satisfaction, to ensure the successful implementation of the ERP system are identified. The results of the exploratory study, carried out on 60 users in 40 Moroccan companies, shows that user satisfaction of ERP is explained by the quality of the ERP system, perceived usefulness and quality of information provided by this type of system. The study also found that the quality engineering change is a predictor of satisfaction measured by user involvement in the implementation of ERP, the quality of communication within such a project and the quality of training given to users.

1 INTRODUCTION

The current context of global economic activity is characterized by a large and permanent competition as well as a large customer requirement for immediate and complex solutions. In this context, process control and continuous improvement become prerequisites for success. As a result, numerous companies around the world are trying to take advantage of an overhaul, using software packages, their information systems, and hundreds of them have opted for systems integrated management ERP (Enterprise Resource Planning) as a basis for the integration of their industrial management (Marbert, Soni & Venkataramanan, 2000).

Companies operate in an environment increasingly complex and changing. They now confront several problems: saturated markets, increased competitiveness, customers more demanding and less loyal, etc. In such an environment, business competitiveness depends increasingly on their flexibility and their ability to innovate, both in their organizational structure, their production as in their mode of exchange with customers and suppliers. However, in their search for competitiveness, the main obstacle faced by companies is the difficulty of obtaining data and accurate information and appropriate interfaces between the various business functions.

This study is interesting on two levels:

- The objective of this article is therefore to identify the drivers of satisfaction of users of ERP systems. On a finer way, we try to determine the satisfaction and enhance the need for good conduct ERP projects to increase the degree of the satisfaction. To do this, it was reduced to build a model for the explanation of this satisfaction.
- In what follows, we will try to review the state of the art in measurement of user satisfaction of IT before submitting the research model and the results of an exploratory study conducted with a sample of 'Moroccan companies.

2 ERP AND ITS CONTRIBUTIONS FOR USERS

The evolution of computing, which is progressing towards greater information sharing and flexibility is

Jalil, F., Zaouia, A. and Bouanani, R.
The Impact of the Implementation of ERP Satisfaction of End Users in Major Moroccan Companies.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 188-195
ISBN: 978-989-758-187-8

a key factor explaining the growing success of ERP to companies. Despite the unquestionable progress they make today, ERP do not fully meet satisfactorily the needs of companies.

2.1 The Emergence of Integrity Management Software

Historically, functional systems businesses were developed on different materials following different methodologies: the achievements are generally heterogeneous both in terms of data representation at the level of processing modes. It follows multiple disadvantages:

- Communication problems between areas expected to share common data;
- Process control challenges due to the multiple treatments required to obtain synthetic statements;
- Students maintenance costs in the absence of modularity resulting low scalability;
- Complexity of the training was the use of very varied software;
- Difficulties for many controllers, in the collection and re-keying data from different systems and serving to consolidate budgets, develop reporting tables, etc.

2.2 What is an ERP?

Acronym of American origin, ERP (Enterprise Resource Planning) is commonly used to designate the integrity management software. The term "ERP" is not totally adequate because it puts only evidence planning appearance. However the French translation "ERP" does not include the planning dimension and its use is problematic.

As defined by Robert Reix (1999), an ERP is a computer application that incorporated the following general characteristics:

- An ERP is a software package: according CXP4, a software package is "a coherent and independent set is service programs, supports, or handling of information and documentation, designed to perform standard IT processes, including the distribution is of a commercial nature and that a user can independently use after installation and limited training.

- An ERP is customizable: standardized product, the ERP is designed originally to meet the needs of various businesses. II usually are different versions by sector (automotive, banking, etc.) and prolonged use. In addition, the adaptation of the product to the needs of a particular business is by setting (choice of management rules, choice of treatment options, choice of data format, etc.). The setting may be accompanied by an appeal has additions of specific programs articulated around standard programs.

- An ERP is modular: it is not a monolithic structure but a set of programs or separable modules each corresponding to a management process: installation and operation can realize autonomously. The division into modules allows you to dial a specific solution for assembly and extend the implementation has different areas of management.

- An ERP is integrated: the various modules are not designed independently they can exchange information according to patterns provided. The PGI guarantees at all times a perfect integrity and data consistency for all users, allowing DC to end interfacing problems, synchronization and double entries.

- An ERP is a management application: it captures the company's transactions (accounting, stock management, order tracking and production program ...) and propagates the information collected to the appropriate levels. However, it contains no optimization program or automatic decision.

2.3 Why Moroccan Businesses They Opt for ERP?

The term ERP comes from the name of the method MRP (Manufacturing Resource Planning), industrial method used since the 1970s for the needs of management and planning of industrial production and computer-aided production management.

As to migration patterns, some companies opt for highly problematic solution to migrate their IT systems through process redesign while others opt for outright deportation of the existing.

Now this type of establishment "Big Bang" is desirable, particularly in the following cases:

 Existence of bottlenecks and sticking points of information between services; the implementation of ERP will provide an opportunity to everyone to overhaul their procedures.

 Excessive heterogeneity of the applications used and abundance of interfaces between business and auxiliary applications with the central accounting system.

 Great difficulty for employees to adapt to new applications when they change service within the company.

In general, it should return to the main benefits and difficulties of ERP to determine whether the context of implementation of ERP is favourable or not.

2.3.1 Organizational Factors

In view of concentration transactions (mergers and acquisitions) facing most industries and growth marked at several international groups, many companies are forced to include in their calendars projects migration to ERP, and, just as the parent company or other subsidiaries.

Moreover, partnership agreements and often require their contractors to implement the same management systems that their reference customers, and to ensure the exchange of data, transparency and reliability of information financial communicated to the third party.

2.3.2 Technical Factors

The evolution of information systems to more sharing, more integration and flexibility are key factors behind the growing success of ERP to companies. Today, as we will examine later, they do not yet fully satisfactory answer to the expectations of the latter.

Nevertheless, they represent the most promising path towards a more comprehensive computer.

2.3.3 Prudential and Internal Control Factors

Indeed, in accounting and financial reporting and management, ERP also provides an audit trail and traceability data, the ability to set up multiple quantitative and qualitative controls to increase assurance of the device internal control during the course of the various transactions through the "workflows" of validation and execution of manual data processing, semi-automatic or automatic.

2.4 The Measurement Models of Successful ERP System

Many models have been developed to evaluate the systems and the success of technology (Davis, 1989b; DeLone and McLean, 2003 1992 Gable et all., 2003; Ifinedo and Nahar, 2006; Sedera and Gable, 2010; Shang Seddon, 2002). These models have been validated empirically by numerous studies in the information system. The results show that many case studies are studied by applying the DeLone & McLean model success using a modeling approach structural equation (Dörr et all., 2013).

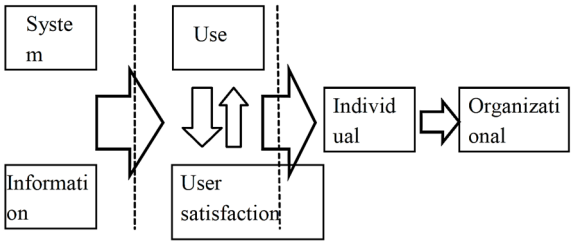


Figure 1: D&M IS Success Model (Delone& McLean.1992).

2.5 Evaluation Approaches

Many researchers have tried to understand the relationship between IT investments and performance, focusing on five main approaches for evaluating IT projects (Bellaaj, 2010). These approaches are:

- Evaluation Approach economic theory (Brynjolfsson): the main objective of this approach is to understand the gap between IT investment and productivity of the organization according to certain economic criteria.
- Evaluation Approach Social Psychology (Davis, 1989a, 1989b; Venkatesh et all., 2003): beyond the economic approach, it incorporates the human factor as a key factor in the process of IT investment and impact assessment.
- Evaluation Approach Based on the analysis of competition: this approach is developed by (Porter and Millar, 1985) explains how technology affects all business. The authors outline the information technology needs to be understood more than just computers, it must be broadly conceived to encompass information that companies create and use as well as a wide range of technologies more increasingly convergent and linked this process the information in their perception of IT they adopt the concept of the value chain to explain the competitive advantages of IT investments.
- Evaluation Method based on strategic alignment: This approach is developed by (Henderson and Venkatraman, 1993), it is widely used by researchers in the information to understand two key concepts system; the first is the adequacy of the information technology goals and strategic objectives of the organization; the second is the functional integration (integration between business and functional areas). This approach suggests that the IT strategy must be consistent with the business strategy to improve organizational performance.
- Evaluation Process Approach: a new conception of assessment is success was brought by this approach based on the theory developed by

emerging process (Markus and Tanis, 2000). This approach highlights the failure of the economic model to assess the success, and proposes a new vision of evaluation not only on the input evaluation on the base, but also based on the use and impacts of IT, by virtue of a valuable creative process.

In this section, we present two examples of evaluation approaches that synthesize the different perspectives of assessment mentioned above. First, we will propose an AHP approach to assessing performance measures ERP (Tsai et all., 2006). Second, we will introduce the Balanced Scorecard approach widely adopted by many researchers to assess the benefits of the ERP system (Chand et all., 2005).

2.6 The Theoretical Foundations

First, we present our conceptual model which is based on both theoretical and empirical background. This framework will be considered a success evaluation model of ERP system that combine causal processes and considerations for evaluating the success of the ERP project in three performance levels: The individual performance, the performance of the task force and performance Organizational (Ifinedo Nahar et al, 2011).

2.6.1 Mathematical Theory of Communication

The mathematical theory of communication (Mason, 1978; Weaver and Shannon, 1949) explains the interaction of three factors: the information system, information such as a product and the impact of information on individual performance and organizationall. This approach is used by (DeLone and McLean, 1992) in their model of success for developing sexual constructions considered the main variable to evaluate the success of the information system.

2.6.2 Innovation Diffusion Theory

Based on the theory of diffusion of innovation, mainly paradigm variables determining the adoption of innovation (Rogers, 1983), three main factors emerged: Innovation /Technological factors, environmental factors and factors Organization. In this taxonomy, each of these factors can be explained in the context of the ERP system. These factors are extremely important in the adoption of ERP phase and they must be integrated in the process of successful ERP system (no success without adopting one hand technologies).

2.6.3 Structuring Theory (AST Approach)

Structuration theory associated with institutional theory Giddens social assessment has been widely applied to understand and explain organizational technology adoption (DeSanctis and Poole, 1994). We focus solely on the AST proposed by DeSanctis and Poole, 1994 to explain how technology brings productivity, efficiency and satisfaction to both individuals and organizations. This approach is based on the school of technology was applied and explained by DeSanctis and Poole, 1994 in their approach to the theory Adaptive Structuring. The ASP is considered a framework to study the variation in the change of the organization and illustrating the impact of advanced technology on organizations. It has been tested on a GDSS (Group Support System to the decision) to answer questions about how technology affects people and organizations that use it, and how it improves the performance of the working group.

3 DEVELOPMENT OF A MODEL SEARCH

Therefore, if a company wants to incorporate an ERP system, even though its operations are not integrated, it should not, alone, buy a software package and associated computer equipment but it is called, also, to acquire know-how and establish a suitable organization of work.

Therefore, methods of effective use of ERP systems require something other than a good computer. Moreover, several companies say they face serious difficulties in the implementation of an ERP system without the technical aspects are actually involved: this is due, in fact, to disregard and neglect human and organizational factors.

Thus, and in support of some researchers the factors considered can be classified keys to the success of an engineering change under the under the following dimensions: the involvement of management Generally, user involvement, communication management, training and the implementation strategy that includes both reengineering business processes (BPR) that the same approach of implementation of these systems.

At the basis of this reasoning, it is assumed that an ERP system is effective at the individual level where its users are satisfied. This level of satisfaction is determined by the quality system implemented in the company, a good quality of the information it provides, high value perceived by

users and good engineering changes necessary for its implementation.

Thus, the various built the model proposed for measuring user satisfaction of an ERP system, detailed below, may be diagrammed as in Figure 2.

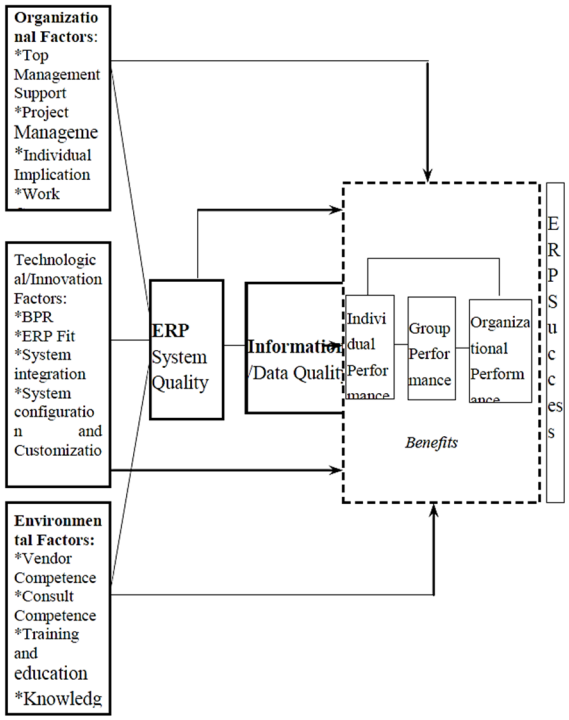


Figure 2: The conceptual model of measuring user satisfaction of an ERP system.

4 DEVELOPMENT HYPOTHESES

This dimension has been used extensively in the literature as the dependent variable success SI. DeLone & McLean (1992) fall within a number of 33 empirical studies published between 1981 and 1987 who enjoy success in terms of user satisfaction (Bailey and Pearson, 1983).

In general, this satisfaction was defined by the attitudes and perceptions. In specifically, this satisfaction was defined as the result of the evaluation that individuals are on continuum "content - dissatisfied"; or the sum of feelings and attitudes towards each of a variety of factors affecting the situation.

4.1 Quality System ERP

This dimension is widely used in the literature (Doll & Torkzadeh, 1988; Davis, 1989; DeLone & McLean, 1992; McGill et all., 1999; etc.). It is a powerful determinant of the effectiveness of IT as well as user satisfaction. The quality of the system relates to the quality of application itself (the different system functionality, ease of use and learning). In addition, it summarizes some issues such as the lack of "bugs" in the system, the user-friendly interface, etc.

Therefore, the hypothesis H1 states: "The better the quality of the system (ERP) is good, more satisfaction is high."

4.2 Quality of Information Provided by the System ERP

The concept of quality of information has been widely used as a key success factor in research in SI. In fact, this construct has been measured primarily by Bailey & Pearson (1983) and Doll & Torkzadeh (1988) as a measure among other satisfaction. This dimension usually includes attributes related to the quality of the information provided by the ERP system, such as the format of the information, clarity of information, accuracy of information, availability of necessary information in real time, the information content, etc.

Therefore, the second hypothesis H2 states: "The better the quality of information provided by the system (ERP) is good, more user satisfaction is high."

4.3 Perceived Utility

This construct is defined by Davis (1989) as the degree to which a person believes that the use of a particular system would increase the work performance. This dimension has been considered as a factor affecting the satisfaction of users that it comprises, on the one hand, items related to the perceived ease of use and, on the other hand, those related to the perceived usefulness. Moreover, Davis (1989) shows that the acceptance of a technology depends on perceptions of users of this technology. Indeed, the Technology Acceptance Model (MAT) assumes two types of beliefs, perceived ease of use and perceived usefulness, determine the intent of the individual who influences the use of technology.

This allows, therefore, bringing forward the third hypothesis H3 namely: "The greater the perceived usefulness by users, the greater their satisfaction is high."

4.4 Quality Engineering Change

As shown above, this new dimension can be

understood by the five under following detailed dimensions.

4.4.1 Involvement

To drive change caused by the implementation of an ERP system, it is essential that this project will become the project of the entire company: from top management to operational:

- **The Involvement of Senior Management**

Indeed, the leaders are not called, only to finance the project but also to take an active role in managing change. This role is mainly to guide the overall operation, encourage local initiative, indicate very clearly the kind of organization that wishes to establish, define the corresponding steps of achievements, etc.

- **The Involvement of Users**

Added to the commitment of senior management and middle management, the implementation of an ERP system can be conducted only by the involvement of the community of operational users and a user project manager full time representing the whole of this community.

However, it is important to note that the involvement of users could not be, in itself, a prerequisite for the proper conduct of change. The latter requires, in addition, good communication management.

4.4.2 Communication

Certainly, the quality of communication within work groups plays an important role in employee attitudes towards change. Where communication and atmosphere were good, new technologies were generally welcomed with enthusiasm, while in groups where members felt compelled to comply with the new rules, reactions were much less favourable. In fact, communication is essential not only to create an understanding and approval of the establishment, but also to win the agreement of users. This communication should begin early, be consistent and continuous.

In addition to good communication during an implementation project of an ERP system, it is inevitable to provide training to users.

4.4.3 Training

Training is seen as an important factor to facilitate change in the organization and introduction of new technologies .This training aims mainly to prepare staff and help them adapt to their new tasks in order to be successful organizational change. It is not intended; only use new systems but also the understanding of new processes and their integration into the system. Hence, training is an ongoing process and updating a challenge.

4.4.4 The Implementation Strategy of an ERP System

The implementation of an ERP system means a continuous learning cycle in which the organizational process supported by ERP systems is aligned gradually with the company's goals. Lequeux (1999) says: "Far from leading a purely IT project, the adoption of ERP should be an opportunity to reconsider the mechanisms and improve the flow participating in the operation of the business, even to consider a business process reengineering".

- **The Business Process Reengineering**

Moreover, the re-engineering of business processes and implementation of ERP systems are inseparable. They should be carried out simultaneously in order to obtain the best fit between the technologies and processes. This adjustment requires considering the role of ERP systems such as infrastructure, which now support the process and no longer functions and, therefore, improve their organizational effectiveness.

- **The ERP System Implementation Approach**

Akkermans and Helden (2001) have focused on ERP systems implementation approach while trying to show that the incremental approach, scalable, based on continuous improvement is a key success factor in the implementation of a project ERP. They add that users of an ERP system are less satisfied if there was a radical approach (Revolutionary) that this approach results in a rigid management style based on a high degree of control and command, Intensive use of external experts, even non staff involvement and therefore a loss of skills and know-how internally.

Thus, and from the previous development on engineering changes, it was agreed to present the hypothesis H4 on this new dimension, "the higher the quality of engineering change is good, more user satisfaction is better ".

This hypothesis derived secondary hypotheses for sub dimensions of engineering change. They are formulated as follows:

- H4a "More DG is involved in the project implementation of an ERP system, more user satisfaction has increased."
- H4b: "More user involvement, the greater their satisfaction is high."
- H4c: "More communication is good, most users are well satisfied."
- H4d: "More training is good; more user satisfaction is very high."
- H4e "The incremental implementation approach can increase user satisfaction more than the radical approach."

5 RESEARCH METHODOLOGY

Once part of the research is defined and the variables of the research are identified, it is important to conduct data collection. For this, a questionnaire, multi-scale, was built and tested with users belonging to both different hierarchical levels as various services, and finally administered face to face in Moroccan companies.

Given that companies have adopted ERP systems are not numerous, it was not possible to focus on a specific industry. The selection of the study population was guided by a single criterion, namely: the existence of an ERP system that is already operating at all levels (all modules are already functional) or at least a good part of the system east.

Data collection has collected a sample of 40 companies surveyed; representing an effective response rate (60.45%). However, it should be noted that the unit of this study is defined as the user of an ERP system. Therefore, the respondent is either the project leader or the leader or one of the senior or middle managers, or one of the last entry clerks. What mattered was the use of the ERP system.

6 RESULTS AND INTERPRETATION

It is important to note that the measurement scales were either adopted from previous work or created for the need of this research.

6.1 Descriptive Analyzes of Research Variables: Evaluation of Measures

After proposing measures to the various concepts identified in the model and collected the data from the selected population, it is appropriate now to ensure the quality of these measures before making adequate statistical treatment. To do this, we made two types of tests for evaluating the measures namely: tests

The dimensionality and reliability test (Cronbach's alpha) (Evrard, Pras & Roux, 1997). Through these purification tests, which are based on principal component analysis ACP was determined for each building its KMO MSA and each of its items.

So we tried to conclude whether built or not is one dimensional and to specify the contribution of each item to the formation of the factor. Finally, we calculated, for each cleared factor, Cronbach's alpha.

6.2 Explanatory Analyzes of Research

Once the measures have been evaluated and the new structures are identified, we proceeded to test hypotheses. This part, devoted to the operationalization of the model and test hypotheses, has identified the following results.

Results thus obtained confirmed the work of DeLone & McLean and those Seddon & Kiew. These results have shown that this satisfaction is explained:

- Primarily by the quality of the system, the quality of information provided by this system and the utility perceived by the users;

- Partially by the quality of engineering changes needed to implement the ERP system. It is true that the data analysis performed could provide only partial verification of this dimension engineering change because, firstly, user involvement, communication and training partially affect that satisfaction on the other hand, the other two sub-dimensions i.e., the involvement of the DG and the implementation strategy does not seem to affect the satisfaction.

Notwithstanding, the results presented are limited to the sample of enterprises and should be interpreted with caution given the nature and sample structure, but also methods of data collection used.

So it will be wise to take this model while increasing the sample size to allow better analysis to improve results. This should be possible since the number of Tunisian companies that are in the process of implementing ERP systems is increasing.

7 CONCLUSIONS

In conclusion, it should be noted that in our time, the information system has become the cornerstone of

consolidating the company's strategy .Thus, the IS manager is asked to provide future solutions enabling the company to be more competitive. It is no longer to increase productivity but to provide the general direction the technological know-how through which the company will be able to adapt its service to the needs of its customers while controlling costs.

Through this article, it is important to note the prominence that ERP systems are currently in Moroccan companies. In fact, these integrated management systems, which are increasingly "backbone" of the SI of the company, need special attention, including in their implementation and evaluation.

Closer to the work of the "Management Information Systems" relating to the determinants of success of IF including the determinants of user satisfaction, the results of this research show that the dimensions outlined in previous studies (Quality System, quality of information and usefulness) remain well determinants of user satisfaction of an ERP system.

REFERENCES

Bellaaj, M., 2010. Technologies de l'information et performance organisationnelle: différentes approches d'évaluation.

Chand, D., Hachey, G., Hunton, J., Owhoso, V. Vasudevan, S., 2005. A balanced scorecard based framework for Assessing the strategic impacts of ERP systems. *Computers in Industry 56, 558-572.*

DeLone, WH, McLean, ER, 1992. Information Systems Success: The Quest for the Dependent Variable. *Information Systems Research 3, 6.-95.*

Davenport, T .., 1998. Putting the Enterprise into the Enterprise System. *Harvard Business Review 76, 121-131.*

Dörr, S., Walther, S., Eymann, T., 2013. Information Systems Success - A Quantitative Literature Review and Comparison. Presented at the *11th International Conference on Wirtschaftsinformatik*, Germany.

Davis, F.D., 1989a. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly 13, 318–340.*

Davis, F.D., 1989b. User acceptance of computer technology: a comparison of two theoretical models. *Management science 35, 982–1003.*

DeSanctis, G., Poole, M.S., 1994. Capturing the Complexity in Advanced Technology Use: Adaptive Structuration Theory. *Organization Science 15, 121–147.*

Henderson, J.C., Venkatraman, N., 1993. Strategic alignement: Leveraging information technology for transforming organizations. *IBM System Journal 38, 472–484.*

Ifinedo, P., 2011. Examining the influences of external expertise and in-house computer/IT knowledge on ERP.

Irani, Z., Sharif, A. Kamal, MM, Love, PED, 2014. Visualising mapping has knowledge of information systems investment evaluation. *Expert Systems with Applications 41, 105-125.*

Irani, Z., Love, P.E.D., 2008. Evaluating information systems Public and private sector. *Elsevier.*

Lequeux, JL, Manager with ERP, integrated management software packages and Internal "Les Editions d'Organisation, Paris, 1999.

Marbert, VA, Soni, A. & Venkataramanan, MA, "An investigation into the ERP in the US industrial companies," *French Industrial Management Review, Vol. 19, N. 4, 2000, pp. 5-13.*

Mason, R.O., 1978. Measuring information output: A communication systems approach. *Information & Management 1, 219–234.*

Markus, M.L., Tanis, C., 2000. the enterprise system experience from adoption to success.

Perotin, P., "Implementation of ERP and organizational integration" *7th Symposium of AIM, Hammamet, 30-1, in June 2002.*

Porter, M., Millar, V., 1985. How Information Gives You Competitive Advantage. *Harvard Business Review 149–160.*

Rogers, E.M., 1983. Diffusion of innovations, 3rd ed. Free Press, United States of America.

Seddon, P. & Kiew, M, "A partial test and development of the DeLone and McLean model of success", Proceedings of the *15th International Conference on Information Systems, December 14-17, 1994 Vancouver, Canada, pp.99-110.*

Stefanou, CJ, C., 2001. A framework for the ex-ante evaluation of ERP software. *European Journal of Information Systems 204-2015.*

Seddon, P., 1997. A Respecification and Extension of the DeLone and McLean Model of IS Success. *Information Systems Research 8, 240–253.*

Tsai, W.-H., Hsu, P.-Y., Cheng, J.M.-S., 2006. An AHP approach to assessing the relative importance weights of ERP performance measures. *International Journal of management & enterprise development 3, 351–375.*

Uwizeyemungu, S., Raymond, L., 2010. Linking the Effects of ERP to Organizational Performance: Development and Initial Validation of an Evaluation Method. *Information Systems Management 27, 25-41.*

Urbach, N., Smolnik, S., 2008. A Methodological *Examination of Empirical Research on Information Systems Success: 2003 to 2007.*

An Evaluation of the Challenges of Multilingualism in Data Warehouse Development

Nedim Dedić and Clare Stanier
Faculty of Computing, Engineering and Sciences, Staffordshire University,
Beaconside, Stafford, Staffordshire, ST18 0AD, U.K.
nedim.dedic@research.staffs.ac.uk, c.stanier@staffs.ac.uk

Keywords: Business Intelligence, Data Warehousing, Multilingualism, Star Schema, Semantic Web.

Abstract: In this paper we discuss Business Intelligence and define what is meant by support for Multilingualism in a Business Intelligence reporting context. We identify support for Multilingualism as a challenging issue which has implications for data warehouse design and reporting performance. Data warehouses are a core component of most Business Intelligence systems and the star schema is the approach most widely used to develop data warehouses and dimensional Data Marts. We discuss the way in which Multilingualism can be supported in the Star Schema and identify that current approaches have serious limitations which include data redundancy and data manipulation, performance and maintenance issues. We propose a new approach to enable the optimal application of multilingualism in Business Intelligence. The proposed approach was found to produce satisfactory results when used in a proof-of-concept environment. Future work will include testing the approach in an enterprise environment.

1 INTRODUCTION

Users today expect to access relevant information in the semantic web in their own language (Gracia et al, 2011). This is also the case when accessing analytical data relevant for decision making using Business Intelligence (BI) applications, such as reports or dashboards. In this paper, we discuss the application of BI in an international context focusing the issue of Multilingualism (ML). Multilingualism is defined further in section two but can be understood as data manipulation and reporting in more than one language. Understanding the issues presented by ML requires discussion of the Data Warehouse (DW), which is a core component of BI (Olszak and Ziemba, 2007). We also discuss data marts and focus on the star schema as the most accepted form of dimensional modelling. We evaluate current solutions and approaches to the implementation of ML using the star schema in BI environment and propose an improved approach.

The rest of this paper is structured as follows: section 2 defines BI and ML and the requirement to support ML in a BI context; section 3 discusses ML in a Data Warehouse environment, outlining DW concepts and design approaches, the role of the star schema and the issues presented when implementing ML in a star schema; section 4 presents the conclusions and evaluation, proposing a revised approach to supporting ML.

2 PROBLEM CONTEXT: BUSINESS INTELLIGENCE AND MULTILINGUALISM

To survive in today's business a company has to continuously improve productivity and efficiency, while management and executives have to make decisions almost immediately to ensure competitiveness (Huff, 2013). Information is used to enable improved decision making and efficiency (Yrjö-Koskinen, 2013; Hannula and Pirttimäki, 2003). This process is supported by activities, processes and applications which are collectively known as Business Intelligence. Business Intelligence was initially used to describe activities and tools associated with the reporting and analysis of data stored in data warehouses (Kimball et al, 2008). Dekkers et al., (2007) define BI as a continuous activity of gathering, processing and analysing data. BI helps companies to out-think the

Dedić, N. and Stanier, C.
An Evaluation of the Challenges of Multilingualism in Data Warehouse Development.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 196-206
ISBN: 978-989-758-187-8

competition through better understanding of the customer base (Brannon, 2010) and can provide competitive advantage (Marchand and Raymond, 2008), and support for strategic decision-making (Popovič et al., 2010). From the early days of computing, computer technology and software has been associated with development in the English language (Hensch, 2005) and Business Intelligence is no exception.

BI is a fast evolving field (Obeidat et al, 2015; Chaudhuri et al., 2011) and although traditional BI focussed on activities such as data warehousing and reporting, the new generation of BI has an additional focus on data exploration and visualisation (Anadiotis, 2013; Obeidat et al, 2015), which increases the demand for ML. The business and legal context of BI is also evolving. With emerging markets and expanding international cooperation, especially in the case of European Union, there is a requirement to support BI in languages other than English. Users today expect to access information in the semantic web in their own language (Gracia et al, 2011). Based on the online profiles of the biggest European companies (Forbes, 2015), most of these companies are international in their nature. Business users expect to be able to use software and applications, including Business Intelligence, in their own language for the purpose of better productivity (Hau and Aparício, 2008). Thus, when expanding their business to new countries, companies need to extend and adopt their BI infrastructure to support optimal use of local languages. In a European context, the requirement for multilingualism has legal underpinnings in some countries. Most European countries have laws on the official use of their respective languages in public communication (Italian Law No. 482, 1999; Constitution of France, 1958; Constitution of Croatia, 1990; Federation Constitution, 1994; Spanish Constitution, 1978). For international companies, this means a requirement to support the use of several languages to obey local laws. Where there is a need to support multiple languages, there is an imperative to enable transfer and processing of textual accessibilities for localization purposes (Vazquez, 2013). This also applies in a Business Intelligence environment.

Multilingualism is complex phenomenon which can be seen from different perspectives and has many definitions (Cenoz, 2013). The European Commission (2008, p.6.) defines it as "the ability of societies, institutions, groups and individuals to engage, on a regular basis, with more than one language in their day-to-day lives". In the Business Intelligence DW and presentation context, we define multilingualism as the ability to store descriptive information and to use this information in a semantic layer in more than one language. The changing attitudes of business users, the importance of emerging and international markets and ever-growing local data warehousing communities are additional issues that justify the application of multilingualism in Business Intelligence. Multilingualism, however, presents challenges for design and reporting in data warehouses.

3 MUTILINGUALISM IN A DATA WAREHOUSE ENVIRONMENT

3.1 Data Warehouse Concepts

Data warehouses are seen as core in the development of BI systems (Olszak and Ziemba, 2007). A widely accepted definition of the Data Warehouse is provided by Inmon (2005) who defined a Data Warehouse as a collection of integrated databases designed to support the DSS (decision support system) function. In this definition, the data warehouse from the architectural point should be almost the same as the source system and may also have data marts, or aggregated tables that are used for reporting and querying purposes. Linstedt et al (2010) propose very similar concept known as the Data Vault approach. Differentiation is only in the context of modelling and storing information inside the Data Warehouse. In the Data Vault approach data is loaded from the source system in its original format (Linstedt et. al, 2010). The Data Vault approach is built around Hubs, Links and Satellites (Jovanović et al., 2014). Hubs represent source system business keys in a master table, links are associations between hubs with validity period, and satellites point to the links containing attributes of transaction with validity period (Orlov, 2014). As the structure of the data is highly normalized (4NF+), this approach for the implementation of a data warehouse is not adequate for direct reporting and requires additional dimensional data marts to enable reporting or querying (Orlov, 2014).

Kimball et al (2008) presented an alternative view of the data warehouse, arguing that a data warehouse should be seen as a collection of data marts, which are used for querying and reporting and connected using conformed dimensions. In this approach, there is no requirement to replicate all the data from the source system, just the data needed by the business.

3.2 The Star Schema

Much of the literature on the development of data warehouses, and particularly the seminal works by Inmon and Kimball, dates from the end of the 20th century/the first decade of the current century. There has been comparatively little recent work on data warehouse design and schema development although there is a significant literature on data warehouse development and optimisation (Cravero and Sepulveda, 2015; Dokeroglu et al., 2014; Sano, 2014; Graefe et al., 2013). Inmon and Kimball are the two seminal figures in this field and although their data warehouse design philosophies, as discussed in section 3.1, are opposed, both propose dimensional modelling and the use of data marts (star or snowflake schema) for reporting (Orlov, 2014). The Data Vault approach introduced by Linstedt (2010) also proposes the use of data marts (in the form of star or snowflake) for reporting. Inmon (1995), Kimball (2008) and Linstedt (2010) all recommend the use of the star schema as the most appropriate design strategy for the development of data marts. Additionally, a survey paper by Sen and Sinha (2005) examined the approaches used by 15 data warehouse vendors and found that 12 of the 15 vendors supported the star schema (alone or in combination with others star schema based approaches).

A star schema is a collection of dimension tables and one or more fact tables (Cios, Pedrycz, Winiarski et al, 2007). Dimensions have a key field and one additional field for every attribute (Kimball et al, 2008; Jensen et al, 2010). The fact table is a central table that contains transactional information and foreign keys to dimensional tables, while dimensional tables contain only master data (Kimball et al, 2008; Jensen et al, 2010; Cios, Pedrycz, Winiarski et al, 2007). In visual model representation, the dimension model resembles a star, thus the name (Jensen, 2010). The main benefits of the star schema design are ease of understanding and a reduction in the number of joins needed to retrieve the data (Cios, Pedrycz, Winiarski et al, 2007).

In dimension tables, the primary key is used to identify the dimensional value, while hierarchy is defined through attributes. Dimension tables do not confirm to the relational model strategy of normalisation and may contain redundancy (Jensen et al, 2010). The Fact table, on the other hand, holds the foreign key to dimensional table values and as there is no redundancy it could be considered to be in 3NF (Jensen et al, 2010). All foreign keys to the dimensional tables build together the primary key for the fact table.

There are other schemas used for the purpose of dimensional modelling, such as snowflake or galaxy. Cios, Pedrycz, Winiarski et al (2007) consider the snowflake and the galaxy schemas as the variations of the star schema, while Inmon (1995), Kimball (2008), Linstedt (2010), Corr and Stagnittno (2014) and Jensen et al (2010) consider it as a separate dimensional modelling philosophy and not as a variation of the star schema.

The star schema is the dimensional modelling concept most used by the industry (Sen and Sinha, 2005). It is recommended as the most appropriate design strategy for development of data marts and is considered as a general dimensional modelling approach in the data warehouse (Inmon, 1995; Kimball, 2008; Linstedt, 2010). Thus, in this paper we focus on the issues of multilingualism exclusively within star schema. Although the accepted design approach for DW development, and regarded as a good fit for business requirements (Purba, 1999), star schemas present issues when handling multilingual systems.

3.3 Multilingualism Issues in Data Warehouse Design

Conventional Business Intelligence uses a process known as ETL [Extract-Transform- Load] (Kimball et al, 2008; Inmon, 2005) in which data is extracted from data source applications, transformed and loaded into a data storage medium, typically a data warehouse or data mart and then analysed to enable meaningful reporting usually via web or desktop applications. However, this process is most effective in a single language environment. Language, including multilingualism, is a difficult issue in software localization (Collins, 2002) especially in a multidimensional context; and the expansion of BI systems to enable reporting in different languages is not trivial. In the context of multilingual websites, several factors have been identified when presenting to different range of audience in different countries

Table 1: Simple Product dimension.

Key	Description	Code	Category	Subcategory	From_Date	To_Date
123	Apples	FA	Fruits vegetables	Fruits	01.01.2014	31012014
124	Beer	DB	Drinks	Alcoholic	01.01.2014	31012014

(Hillier 2003). The most important are cultural context and accessibility of applications in local language. Creating and maintaining a web environment in a multilingual perspective creates special challenges, both cultural and technical (Huang and Tilley, 2001). Besides the standard issues that arise during translation process from one to other language (such as meaning of words, terminology, phrases, text direction, date formats, etc.) we face additional technical issues when translating texts in computer-based environment (Hillier, 2003).

These problems may range from different application environments to different implementation standards. To optimally apply multilingualism to existing Business Intelligence environment it is necessary to identify the issues in a BI environment. The next section examines three possible workarounds based on amendments to the star schema: including additional attributes, extending the primary key and providing additional dimension tables. All these solutions, as discussed below introduce problems such as extreme data redundancies leading to performance issues, and implementation and maintenance difficulties.

3.3.1 Adding Additional Attributes for New Languages to Dimension

One approach, derived from Kimball's method for delivering country-specific calendars (Kimball and Ross, 2011), recommends that where there are new values for the dimension tables in star schema, we simply add new attributes to the dimension. This approach is also proposed by Imhoff et al (2003) as a solution for simultaneous bilingual reporting. Imhoff et al (2003) state that if we need to provide the ability to report in two or more languages within the same query, we need to publish the data with multiple languages in the same row. When implementing dimensions using this approach, attributes should be descriptive, added in the form of textual labels that consist full words, without missing values, discreetly valued and quality assured (Kimball et al 2008). This is illustrated by the simple Product dimension shown Table 1. As we see, attributes in the Product dimension (Description, Code, Category and Subcategory) are textual fields and at this point, the star schema would be sufficient for the most companies to implement functional and optimal data marts. The limitation of this approach in a multilingual environment would be extremely large dimension tables. For example, if there are ten descriptive attributes for "product" dimension, in the case of the five languages, there would be an additional forty columns.

To demonstrate the problem, we convert the product dimension table (Table 1) to a conceptual view (Table 2a). The sample Product dimension, based on Table 2a, which additionally includes German, Italian and Bosnian language besides English would look like Table 2b.

Table 2a: Conceptual view of the Product dimension.

Key *(Primary Key)*
Description
Code
Category
Subcategory
From_Date
To_Date

Table 2b: Product dimension in English, German, Italian and Bosnian language.

Key *(Primary Key)*
Description
Code
Category
Subcategory
From_Date
To_Date
Description_DE
Code_DE
Category_DE
Subcategory_DE
Description_IT
Code_IT
Category_IT
Subcategory_IT
Description_BA
Code_BA
Category_BA
Subcategory_BA

As we see, new attribute columns for German, Italian and Bosnian language are added for every possible textual description. They have suffix **_DE**, **_IT** and **_BA** (Table 2b).This simplified example does not fully convey the scale of the problem. In implementation practice, the Product dimension might contain more than 20 of textual attributes and the problem would be replicated for all dimensions. A real-world example of a Product dimension would include descriptive attributes to be used as reporting aggregates, for example, description, category, subcategory, assortment, assortment area, buying department, brand, brand origin, country, international categorization, product level, season information, product state, class and type.

As they require large amounts of maintenance

time and CPU (Poolet, 2008), large and wide dimension tables can be problematic, especially for rapidly changing dimensions such as a Customer dimension (Ponniah, 2004). Rapidly changing dimensions are those dimensions where attribute or hierarchical values change frequently (Boakye, 2012). As an example, consider a Customer dimension with several million rows of data intended to be used in five languages. In this example, the Customer dimension has descriptive attributes such as "buying category", "buying frequency" and "monetary value" in all five languages. These categories are intended to be updated on a daily basis. This and similar scenarios could lead to system overhead on a daily basis. In addition, wide dimension tables require duplicate storage for descriptive attributes and make ETL transformation complex as the language-based columns must be taken into account. More complex SQL/MDX statements are required with different language-based columns to change the language of data previews at the semantic level (reports, queries or dashboards). Moreover, queries that return data sets must be re-executed in the required language. There are other external, but related problems caused by using this approach. For example, updating or changing some descriptive hierarchical attributes that are used as basis for tables containing aggregated data. Suppose a specific group of products change their category from non-alcoholic drinks to energy drinks, affecting also subcategories. It is necessary to update the dimension table to change the descriptive records for every language and also to re-aggregate the data in tables holding aggregated data. In this scenario, we would have to delete all data in tables that hold aggregate data according to category and re-aggregate. The process of re-aggregation could take several days if we have billions of records in fact tables, which is not unusual; Wallmart.com sells more than 4.000.000 and Amazon.com more than 350.000.000 different products (Scrapehero.com, 2015). The situation is more critical with wide dimension tables that represent rapidly changing dimensions. The overhead would also increase as the company needs to store more languages.

3.3.2 Extending a Primary Key to Include Language Identifier

The second approach, discussed by Imhoff (2003), proposes extending a primary key to include a language identifier. As we can see from Table 3, the limitation in this case is duplication of the records with every new language. With five languages for the product dimension, which for example holds one million of data, there would be five million records.

Larger dimension tables slow the process of query execution and make it harder to manage updates according to the slowly changing dimension rules. Slowly changing dimensions are dimensions whose attribute or hierarchical values change over time, but unlike rapidly changing dimensions, values are changed unpredictably and less frequently (Kimball et al, 2008). This approach to multilingualism is problematic also for rapidly changing dimensions (Ponniah, 2004), and as with the additional attributes approach, makes heavy increased demands in terms of maintenance time and CPU (Poolet, 2008). From a memory perspective this approach is less efficient than that discussed in 3.3.1 as it doubles storage requirements with every additional language. This approach also suffers from the semantic layerproblems previously discussed: to change the language of data preview on semantic layer (reports, dashboards), query statements that return data sets must be re-executed. This approach, unlike the additional attributes approach, does not lead to more complex ETL transformations and SQL/MDX statements but the same problems exist with regard to rapidly changing dimensions and changing the structure of externally aggregated tables. For companies using several languages and holding millions of records in their dimensions, re-executing queries and re-aggregating data according to a specific language can be time, memory and CPU demanding.

3.3.3 Schema and Multiple Dimensional Tables Solution

A third approach discussed by Kimball (2001), Imhoff et al (2003) and Corr and Stagnittno (2014), proposes implementing one fact table and multiple dimensional tables – depending on the number of languages required. Different languages are saved in different database schema and/or in different tables. For example, for five different languages, we would have five "product" dimension tables - one for every language. For the same example, if there are one hundred initial dimensions in data warehouse, five hundred dimension tables would be required to satisfy the MLrequirements for five languages. This approach has numerous limitations. As additional tables and possibly additional schemas are needed in the data warehouse, this approach makes ETL processes more complex as the language-based tables must be planned for. It requires additional transformations to every table for every additional

Table 3: Product dimension with extended primary key.

Key	Lang	Description	Code	Category	Subcategory	From Date	To Date
123	EN	Apples	FA	Fruits vegetables	Fruits	01.01.2014	31.01.2014
124	EN	Beer	DB	Drinks	Alcoholic	01.01.2014	31.01.2014
123	DE	Äpfel	FA	Obst und Gemüse	Obst	01.01.2014	31.01.2014
124	DE	Bier	DB	Getränke	Alcoholisch	01.01.2014	31.01.2014
123	IT	Mele	FA	Frutta e Verdura	Frutta	01.01.2014	31.01.2014
124	IT	Birra	DB	Beve	Alcolico	01.01.2014	31.01.2014
123	SI	Jabloka	FA	Sadje in Zelenjava	Sadje	01.01.2014	31.01.2014
124	SI	Pivo	DB	Pijače	Alkoholna	01.01.2014	31.01.2014
123	BA	Jabuka	FA	Voće i povrće	Voće	01.01.2014	31.01.2014
124	BA	Pivo	DB	Pića	Alkoholna	01.01.2014	31.01.2014

language. The data to be used for aggregation and reporting is doubled and so is the metadata for tables and schemas. This approach requires more complex SQL/MDX statements than two previous approaches, and changing the language of data preview at the semantic level requires the query to be re-executed. Changing any descriptive data in dimensions requires re-aggregation of relevant tables holding aggregated data, which can be critical considering the ETL and SQL/MDX complexity of this approach. If one part of the business (country), for example, changes the ID for a specific dimension value, this could lead to consistency problems. Having different IDs for the same data category in different languages disables consolidated reporting for that aspect at the enterprise level. Other subtle problems that might arise when using this approach as discussed by Kimball (2001) include: the possibility of translating two distinct attributes as the same word in a new language causing ETL and reporting problems; reports cannot preserve sort orders easily across different languages. To overcome these problems, this concept requires additional programming or the application of additional or surrogate keys as actual keys in fact table.

3.3.4 Discussion

Besides data redundancy, sub-optimal memory usage and complex management, the common limitation of the all three approaches is the fact that the business users need to re-execute the query or report to change the interface language. As BI-based reports and queries do complex calculations besides querying huge amounts of data from the data warehouse, this is problematic and introduces performance issues. Further, in the case of translation corrections for master data, corrections have to go through the whole ETL process. For example, to change an attribute description for a specific language, it is necessary to change the value in the source system, then wait for the execution of the process for the respective master data. Other options, such as normalizing star schema dimensions for the purpose of multilingualism could lead to the Star schema developing into a snowflake schema. A snowflake schema is not an optimal design solution for data marts with large amounts of data. Issues include: the execution of the main query needs to consider all joins between tables in the same dimension; multi-level dimension tables inside one dimension have to be joined during query execution; the structure of the snowflake is complex and includes large number of database tables per dimension; harder implementation and maintenance of ETL processes; greater complexity in reorganization than the star schema; changes in the semantic model could lead to the extensive reorganization which would use additional resources.

3.4 Vendor Specific Approaches: SAP Extended Star Schema

We reviewed the data warehouse and BI software market and found that the biggest vendors, such as Oracle, IBM and Microsoft, support one or more of three approaches explained above. However, one of the biggest vendors in Business Intelligence, SAP proposes a specific solution for ML, using the concept of an extended star schema, which also includes language as part of the key. In this approach the dimensions and the fact table are linked to one another using abstract identification numbers (dimension IDs), which are contained in the key part of the respective database table (SAP, 2015). Dimensions are not represented as one table with redundant data as in classical star schema. Values from the tables that hold information about a specific dimension attribute text or value are mapped to an abstract dimension key.

This is an implementation driven approach which is only supported by SAP BW. This means it cannot be seen as a general design solution as it is a vendor

specific proprietary solution which relies on complex joins to retrieve content for reporting purposes.

4 PROPOSED SOLUTION AND FUTURE WORK

4.1 Conclusions

We have reviewed the problems presented by ML in a BI context and discussed the limitations of current solutions to the design challenge of supporting ML in the star schema. Existing solutions propose design workarounds based on extensions of the star schema but lack theoretical underpinnings and introduce data redundancy and data manipulation and performance and maintenance issues. The challenge of ML in a data warehouse environment is to develop an approach that can support ML without introducing inefficiencies into the star schema.

4.2 Proposed Solution

Imhoff et al (2003) propose that language-based values should be generated during the delivery process. Corr and Stagnittno (2014) suggest a similar approach and propose attribute name translation handled by the BI tool semantic layer. The main challenge to providing support for ML in the context of the Star Schema is that attribute and hierarchy descriptions are saved inside the dimensional tables. We therefore propose an alternative approach that would regard the Star Schema as a higher level entity and save textual descriptions from attributes and hierarchies elsewhere as language files.

Figure 1 shows the concept of proposed approach based on a three-layered BI architecture idea. We describe more detailed technical functionality of the proposed solution in Figure 2. Figure 1 shows that the master and transactional data are extracted from source layer applications into a staging area. From the staging area data can be extracted directly into reporting Data Marts, following the philosophy of Kimball (dashed arrows), or to Data Warehouse and then to reporting Data Marts, following the Data Vault and Inmon philosophies (full line arrows). Before storing data into reporting Data Marts, attributes and hierarchical descriptions are extracted, together with their IDs, to language files elsewhere on the server. As we extract attributes and hierarchical descriptions and their IDs to separate language files, we store only integer values to dimensional tables. Descriptions of attributes and hierarchies would be associated with relevant IDs from the dimensional tables during report or query execution (on the fly), depending on the default language or language selected. This concept has numerous benefits compared with conventional methods of enabling multilingualism in Business Intelligence environment. Working only with integers makes SQL/MDX queries faster, reduces table size and storage requirements (Heflin and Pan, 2010; Smith, 2012).

Initially, the end user sees an *Initial Filter Page* (ivp.file) that enables filtering of the content from data marts to be presented in the *Reporting Page* (rep.file). *Initial Filter Page* uses a default language set-up via *Global Variables* and *Language Content Pages* (steps a, b, c and d). The application sends an SQL/MDX request defined using *Initial Filter Page* to the data mart (step 1). The first results set is than sent to *Check and Define Page* (cdp.file), which takes it and defines attribute and hierarchical numerical values from dimensions as language variables (step 2). The second result with variabilized values is then sent to *Reporting Page* (rep.file) (step 3). *Reporting Page* communicate with *Global Variables* and *Language Content Page* using header (steps 4 and b), thus it reads the relevant configuration settings for the language before taking the second result set. After taking the second result set, *Reporting Page* reads language values for attributes and hierarchies from *Language Content files*, assigns them and displays final Result Set in the form of business readable report (steps 5 and c).

Using this approach, there is no need to re-execute the SQL/MDX query to change the preview language – we need only to read new language values from the *Language Content files*. As there is no need for new query execution, switching between languages would be easy and fast.

The manipulation of the language content would be easier for business users, as this could be supported by additional tools or modulesthat enable manipulation of textual descriptions saved in textual files. In this context, we consider the idea from Arefin, Marimoto and Yasmin (2011) that efficient Content Management System (CMS), or better Multilingual CMS (MCMS), would help us to overcome the technical limitations of multilingual content in semantic web. In the proposed solution, language files are part of the Warehousing layer, but content is manageable by MCSM.

Adding new languages for semantic web interface would not be dependent on the languages that exist in the source system. If MCSM is implemented so that it has a backend and a frontend for example, we could

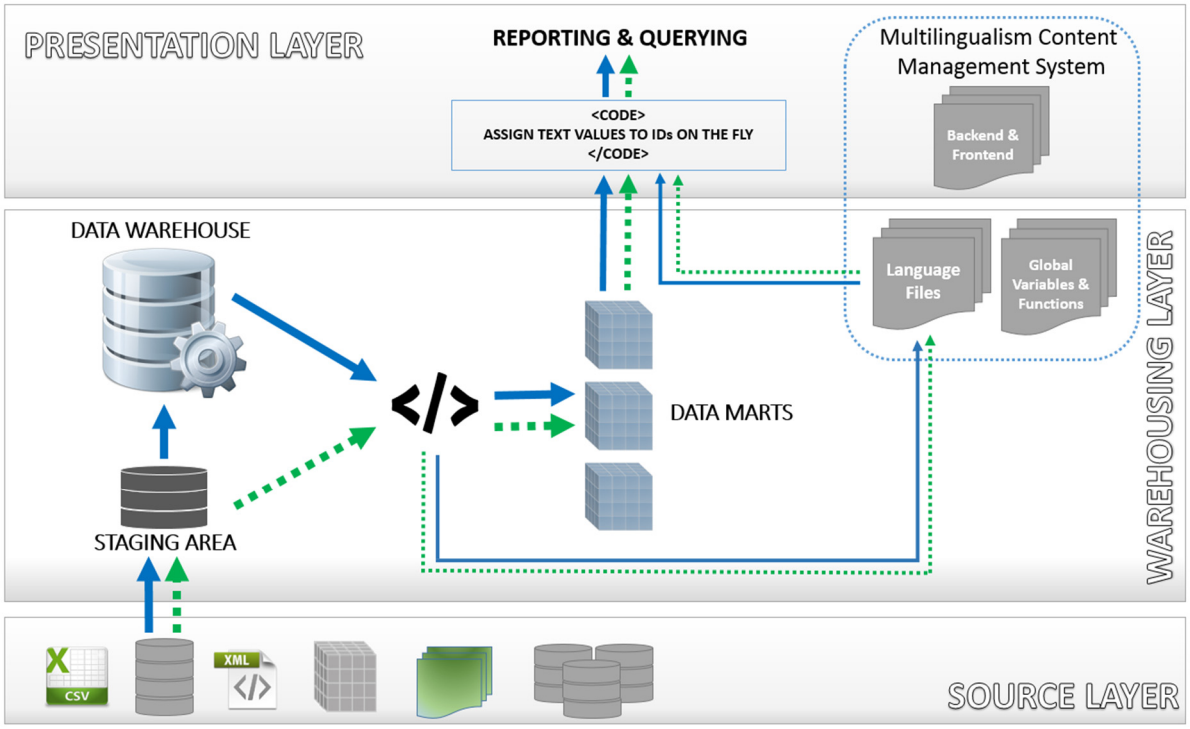


Figure 1: The concept underlying the proposed approach based on three-layered BI architecture.

define new languages through the MCSM backend. This would allow the business user to change descriptions for languages as required.

As it would be possible for business users to change descriptive content directly and by themselves, the ETL process required to perform language changes would be simplified or in some cases eliminated. In conventional solutions, the whole ETL process may be performed just to change a small descriptive value for the specified master data. This is unnecessary in our proposed solution although it would be highly recommended that values should be changed in source systems as well to reflect corrections made in language files. Otherwise, if there is a future need to load whole master data for specific dimension, it would overwrite the corrections made. We recommend using language files only for smaller language corrections and executing standard ETL process when dealing with three or more corrections at the same time.

This approach simplifies the manipulation of textual changes which would be performed outside the values stored in dimensions. This is especially important with dimensions which are typically large such as Article, Product or Customer as there would be no need to re-aggregate existing data according to textual descriptions of attributes or hierarchies. This would allow a more optimal use of database memory as there would be no duplicated data descriptions in dimensions.

Our proposed approach was implemented in a proof-of-concept (PoC) artefact. We had one fact table for Sales data, one dimension table for Product Categories data, and one language file holding Product Categories data.The fact table had Product Category ID as primary key and Price and Amount as measures. Product Category dimension had Product Category ID used for relation with fact table and a Description field holding descriptions of categories. The language file had the same descriptive data as the Product Category dimension. The PoC was tested with 107.768 records in the fact table, 97 records in Product Category dimension table and 97 records for Product categories as language file. For testing purposes, we wanted to show total price per category in our report. In our first PoC method, we multiplied Amount and Price from fact table per Product Category ID and assigned descriptions from language file during report execution - to present the names (descriptions) of categories shown. Using a second method we made the same multiplication, however, we used the names (descriptions) of categories from Product Category dimension and assigned them result set using SQL JOIN.

Results from the PoC showed enviable improvement in performance when using language

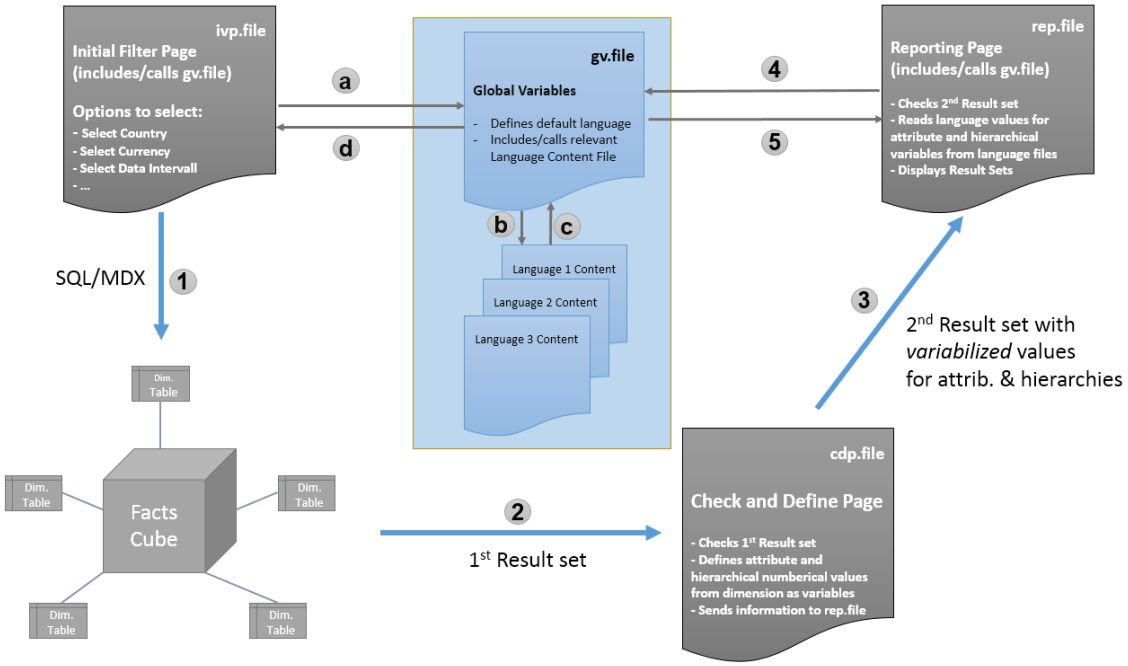


Figure 2: Detailed description of technical functionality of the proposed approach.

files. The average execution time for a report that sums the product sales for 97 different categories using language files method was seven times faster than using conventional methods of implementing star schema. We executed PoC report 228 times: 139 times for method based on language files and 89 times for other method. Using the language files approach we could perform smaller descriptive language changes quickly (simultaneously in data mart and language files). The approach was also less expensive in terms of memory. As this approach can be implemented in the form of add-on to existing monolingual DW structure, it would be optimal way to enable multilingualism in existing BI environment.

4.3 Future Work

The findings from the PoC artefact were encouraging and provide the basis for our future work which will test our solution in a real world environment, as part of a MCSM. Future challenges include integrating multilingualism into the Business Intelligence framework and addressing migration issues as businesses move from single language to multilingual business intelligence systems.

REFERENCES

Anadiotis, G., 2013. Agile business intelligence: reshaping the landscape. , p.3.

Arefin, M.S., Morimoto, Y. & Yasmin, A., 2011. Multilingual Content Management in Web Environment. In *2011 International Conference on Information Science and Applications*. pp. 1–9.

Boakye, E.A., 2012. From Design and Build to Implementation and Validation. In C. McKinney, ed. *Implementing Business Intelligence in Your Healthcare Organization*. Chicago: HIMSS, pp. 73–86.

Brannon, N., 2010. Business Intelligence and E-Discovery. *Intellectual Property & Technology Law Journal*, 22(7), pp.1–5.

Cenoz, J., 2013. Defining Multilingualism. *Annual Review of Applied Linguistics*, 33, pp.3–18.

Chaudhuri, S., Dayal, U. & Narasayya, V., 2011. An overview of business intelligence technology. *Communications of the ACM*, 55(8), p. 88–98 .

Cios, K.J. et al., 2007. *Data Mining: A Knowledge Discovery Approach* 1st ed., New York, USA: Springer Science+Business Media, LLC.

Collins, R.W., 2002. Software localization for internet software: Issues and methods. *IEEE Software*.

Corr, L. & Stagnittno, J., 2014. *Agile Data Warehouse Design: Collaborative Dimensional Modeling, from Whiteboard to Star Schema*, Leeds: DecisionOne Press.

Cravero, A. & Sepúlveda, S., 2015. Using GORE in Data Warehouse: A Systematic Mapping Study . *Latin America Transactions, IEEE (Revista IEEE America Latina)* , 13(5), pp.1654 – 1660.

Croatian Government, 1990. The Constitution of the Republic of Croatia. , p.Article 12.1. .

Dekkers, J., Versendaal, J. & Batenburg, R., 2007. Organising for Business Intelligence: A framework for

aligning the use and development of information. In *BLED 2007 Proceedings*. Bled, pp. 625 – 636.

Dokeroglu, T., Sert, S.A. & Cinar, M.S., 2014. Evolutionary Multiobjective Query Workload Optimization of Cloud Data Warehouses . *The Scientific World*, 2014(14), pp.1 – 16.

European Comission, 2008. *Final Report: Commission of the European Communities High Level Group on Multilingualism*, Luxembourg.

Forbes, 2015. The World's Biggest Public Companies. *Forbes.com*. Available at: http://www.forbes.com/global2000/list/ (Accessed November 24, 2015).

Gouvernement de la République française, 1958. Constitution of France. , p.Article 2.1. .

Government of Federation of Bosnia and Herzegovina, 1995. Federation Constitution. , p.Article 6.1.

Gracia, J. et al., 2012. Challenges for the multilingual Web of Data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 11, pp.63–71.

Graefe, G. et al., 2013. Elasticity in cloud databases and their query processing. *International Journal of Data Warehousing and Mining*, 9(2), p.1.

Hannula, M. & Pirttimäki, V., 2003. Business intelligence empirical study on the top 50 Finnish companies. *Journal of American Academy of Business*, 2, pp.593–599.

Hau, E. & Aparício, M., 2008. Software Internationalization and Localization in Web Based ERP. In *SIGDOC '08 Proceedings of the 26th annual ACM international conference on Design of communication*. New York: ACM , pp. 175 – 180.

Heflin, J. & Pan, Z., 2010. Semantic Integration: The Hawkeye Approach. In *Semantic Computing*. Haboken, New Jersey, US: IEEE Press; John Wiley & Sons, pp. 199–227.

Hensch, K., 2005. IBM History of Far Eastern Languages in Computing, Part 1: Requirements and Initial Phonetic Product Solutions in the 1960s. *IEEE Annals of the History of Computing*, 27(1), pp.17–26. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1401743.

Hillier, M., 2003. The role of cultural context in multilingual website usability. *Electronic Commerce Research and Applications*, 2(1), pp.2–14. Available at: www.elsevier.com.

Huang, S. & Tilley, S., 2001. ssues of content and structure for a multilingual web site. In *Proceedings of the 19th annual international conference on computer documentation*. ACM, pp. 103–110.

Huff, A., 2013. Big Data II: Business Intelligence - Fleets Analyze Information from several sources to improve overall performance. *Commercial Carrier Journal*, (April), p.53.

Imhoff, C., Galemmo, N. & Geiger, J.G., 2003. *Mastering Data Warehouse Design: Relational and Dimensional Techniques*, Indianapolis: Wiley Publishing, Inc.

Inmon, B.W., 2005. *Building the Data Warehouse* 4th ed., Indianapolis: John Wiley & Sons.

Inmon, B.W., 1995. *Building the Data Warehouse* 1st ed., New York: Wiley.

Jensen, C.S., Pedersen, T.B. & Thomsen, C., 2010. *Multidimensional Databases and Data Warehousing* 1st ed., Morgan & Claypool Publishers.

Jovanovic, V., Subotic, D. & Mrdalj, S., 2014. Data Modeling Styles in Data Warehousing. In *37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. Opatija, Croatia: MIPRO, pp. 1458 – 1463.

Kimball, R., 2001. Design Tip #24: Designing Dimensional In A Multinational Data Warehouse. *Kimball Group*. Available at: http://www.kimballgroup.com/2001/06/ design-tip-24-designing-dimensional-in-a-multinational-data-warehouse/ Accessed December 1, 2015).

Kimball, R. et al., 2008. *The Data Warehouse Lifecycle Toolkit* 2nd ed., Indianapolis: John Wiley & Sons.

Kimball, R. & Ross, M., 2011. *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling* 2nd ed., John Wiley & Sons.

Kingdom of Spain, 1978. Spanish Constitution. , p. Article 3.1.

Law No. 482, 1999. Norme in materia di tutela delle minoranze linguistiche storiche. *Gazzetta Ufficiale*, 297 (Article 1.1).

Linstedt, D., Graziano, K. & Hultgren, H., 2010. *The Business of Data Vault Modeling* 2nd ., Lulu.com.

Marchand, M. & Raymond, L., 2008. Researching performance measurement systems: An information systems perspective. *International Journal of Operations & Production Management*, 28(7), pp.663 – 686.

Obeidat, M. et al., 2015. Business Intelligence Technology, Applications, and Trends. *International Management Review*, 11(2), pp.47–56.

Olszak, C. & Ziemba, E., 2007. Approach to Building and Implementing Business Intelligence Systems. *Interdisciplinary Journal of Information, Knowledge & Management*, 2, pp.135–148.

Orlov, V., 2014. Data Warehouse Architecture: Inmon CIF, Kimball Dimensional or Linstedt Data Vault? *WMP Blog*. Available at: http://blog.westmonroepartners.com/data-warehouse-architecture-inmon-cif-kimball-dimensional-or-linstedt-data-vault/ (Accessed February 20, 2015).

Ponniah, P., 2004. *Data Warehousing Fundamentals: A Comprehensive Guide for IT Professionals*, New Yor: John Wiley & Sons.

Poolet, M.A., 2008. Data Warehousing: Rapidly Changing Monster Dimensions. *SQL Server Pro*. Available at: http:// sqlmag.com/database-administration/data-warehousing-rapidly-changing-monster-dimensions (Accessed November 24, 2015).

Popovič, A., Turk, T. & Jaklič, J., 2010. Conceptual Model of Business Value of Business Intelligence Systems. *Management: Journal of Contemporary Management*, 15(1), pp.5–29.

Purba, S., 1999. *Handbook of Data Management* 3rd ed., Boca Raton, FL, US: Auerbach, CRC Press.

Sano, M. Di, 2014. Business Intelligence as a Service : a new approach to manage business processes in the

Cloud. In *2014 IEEE 23rd International WETICE Conference*. Parma, pp. 155–160.

SAP AG, 2015. SAP Library - XML: BW - Data Warehousing - Modeling. Available at: http://help.sap.com/saphelp_snc70/helpdata/en/4c/89dc37c7f2d67ae10000009b38f889/frameset.htm (Accessed March 2, 2015).

Scrapehero, 2015. How many products does Walmart.com sell in comparison to Amazon.com? *Scrapehero.com*. Available at: http://learn.scrapehero.com/how-many-products-does-walmart-com-sell-vs-amazon-com/ [Accessed November 24, 2015].

Sen, A. & Sinha, A.P., 2005. A comparison of data warehousing methodologies. *Communications of the ACM*, 48(3), pp.79–84.

Smith, P., 2012. *Professional Website Performance: Optimizing the Front-End and Back-End*, Indianapolis, USA: John Wiley & Sons.

Vázquez, S.R., 2013. Localizing Accessibility of Text Alternatives for Visual Content in Multilingual Websites. In *ACM SIGACCESS Accessibility and Computing*. ACM, pp. 34–37.

Yrjö-Koskinen, P., 1973. Johto – Tiedontarve – Tiedonhankinta. Miten informaatiopalvelu voi palvel la yrityksen johtoa? *Publications of Insinöörijärjestö*, No.77 - 73.

Towards Keyword-based Pull Recommendation Systems

María del Carmen Rodríguez-Hernández[1], Sergio Ilarri[1], Raquel Trillo-Lado[1] and Francesco Guerra[2]
[1]*Department of Computer Science and Systems Engineering, University of Zaragoza, Zaragoza, Spain*
[2]*University of Modena and Reggio Emilia, Modena, Italy*
{*692383, silarri, raqueltl*}*@unizar.es, francesco.guerra@unimore.it*

Keywords: Keyword-based Search, Recommendation Systems, Mobile Computing, Hidden Markov Model, Information Retrieval.

Abstract: Due to the high availability of data, users are frequently overloaded with a huge amount of alternatives when they need to choose a particular item. This has motivated an increased interest in research on recommendation systems, which filter the options and provide users with suggestions about specific elements (e.g., movies, restaurants, hotels, books, etc.) that are estimated to be potentially relevant for the user.
In this paper, we describe and evaluate two possible solutions to the problem of identification of the type of item (e.g., music, movie, book, etc.) that the user specifies in a pull-based recommendation (i.e., recommendation about certain types of items that are explicitly requested by the user). We evaluate two alternative solutions: one based on the use of the Hidden Markov Model and another one exploiting Information Retrieval techniques. Comparing both proposals experimentally, we can observe that the Hidden Markov Model performs generally better than the Information Retrieval technique in our preliminary experimental setup.

1 INTRODUCTION

Recommender systems (Jannach et al., 2010; Kantor et al., 2011; Adomavicius and Tuzhilin, 2005) suggest (relevant) items to users. The suggestions can help to solve certain decision-making problems which are presented to the users, such as which books to buy, which movies to watch, or which online news to read. They try to adapt the suggestions to each user individually, based on his/her preferences.

Existing pull-based (reactive) recommendation approaches usually assume that the type of item needed by the user is accurately determined by using some external procedure. For example, the user could select an option from a list of predefined types of items. Although this direct selection is very precise and it may be practical in some contexts, we argue that this approach lacks generality and is quite rigid for the user. For example, in a dynamic environment where new data sources could appear or disappear at any time, it may be inconvenient or difficult to have a predefined set of available types of items collected in a static list of options. Moreover, a solution based on a selection among a list of options could be tedious and uncomfortable for the user, who may be forced to use a specific vocabulary and patiently navigate menus.

Therefore, we advocate offering a keyword-based interface to allow users to freely express their needs. For the detection of the type of item requested by a user we consider the existence of a database that contains information about the different types of items available, according to the Entity-Relationship (E/R) schema shown in Figure 1. A part of the schema ("item datasets") focuses on the items available: an item may have a type and can be described through a list of features (name-value pairs). Another part of the schema ("ratings") stores information about the ratings of the items: for each combination user-context-item we may have a specific rating (if available), and each context is characterized by a set of variables. By associating context information to each rating, the E/R schema acknowledges that the context of the user has an impact on the user's perception of the usefulness of different items, as advocated by the so-called *Context-Aware Recommendation Systems* (*CARS*) (Adomavicius and Tuzhilin, 2011); a typical influential context variable is the location of the user (Levandoski et al., 2012). The double rectangles in the E/R diagram indicate weak entity types (Teorey et al., 1986) and the diamonds are oriented towards the regular entity type/s they depend on. In this paper, we focus only on the "item datasets" fragment of the E/R schema.

While intensive research has been performed in the area of keyword-based searching, the use of keyword-based systems as a support for recommen-

Rodríguez-Hernández, M., Ilarri, S., Trillo-Lado, R. and Guerra, F.
Towards Keyword-based Pull Recommendation Systems.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 207-214
ISBN: 978-989-758-187-8

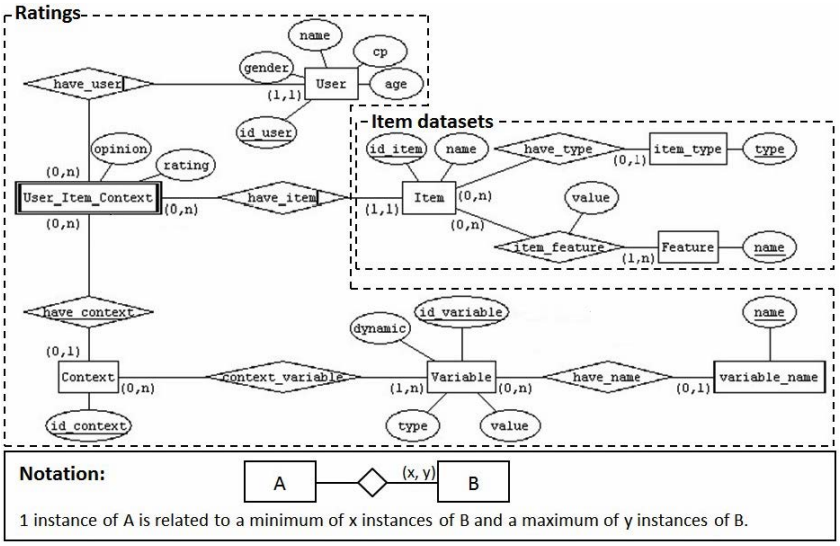


Figure 1: Entity-Relationship diagram modeling data for a pull-based context-aware recommendation system.

dation systems is rather scarce. We believe that keyword-based searching approaches for relational databases cannot be directly applied to help users define their interests for a recommendation system. The reason is that those approaches focus on a different problem; for example, techniques such as those proposed in (Bergamaschi et al., 2010; Bergamaschi et al., 2013) are specialized in queries that retrieve information from several tables at the same time (i.e., join queries), and therefore their adaptation or direct application in scenarios where simpler queries are often needed may result in inefficient solutions. In order to understand the recommendation needs of the user by using keywords, the recommendation system must be able to interpret the semantic meaning of the keywords typed by the user and identify the type of item to recommend. For this purpose, it might need to take into account semantic relations such as synonyms, similar or related keywords (e.g., restaurant – hungry, coffee shop – bar – sleepy), etc.

The current goal of this paper is to start exploring possible approaches for the identification of the type of item requested by a user in a pull-based recommendation process (del Carmen Rodríguez-Hernández and Ilarri, 2015), by using keywords in the user request. For example, if a user introduces in the system the keywords "place to eat" the system must be able to interpret that the user is searching items of the type "restaurant" (or similar types of items, like "bars", if available) without the need to choose the item type from a list previously defined in the system. In this paper, two alternative methods are considered: a solution based on the Hidden Markov Model (HMM) (Rabiner, 1989) and a solution based on the application of traditional Information Retrieval (IR) techniques (Salton and McGill, 1986).

The structure of the rest of this paper is as follows. Section 2 discusses some related work. In Section 3 and 4, we present the HMM and IR approaches, respectively. In Section 5, a set of experiments is conducted to evaluate both proposals. Finally, we conclude the paper and present some lines of future work in Section 6.

2 RELATED WORK

In the area of Information Retrieval (IR) (Salton and McGill, 1986), where generally the data are unstructured (e.g., searching relevant documents in the Web), the problem of keyword-based query answering by using an inverted index (Zobel and Moffat, 2006) has been studied. For structured data, the field of keyword-based search has started to emerge more recently (Chakrabarti et al., 2010). There are several systems that support keyword-based searching over structured data sources, such as BANKS (Aditya et al., 2002), DBXplorer (Agrawal et al., 2002), DISCOVER (Hristidis and Papakonstantinou, 2002), KEYRY (Bergamaschi et al., 2011), QUEST (Bergamaschi et al., 2013), and KEYMANTIC (Bergamaschi et al., 2010). As an example, EASE (Li et al., 2008) is a generic keyword search method which allows to index and query large collections of heterogeneous data (unstructured, semi-structured, and structured data). The authors of EASE extended the tradi-tional inverted index in order to provide keyword-based search, and additionally they proposed a novel

ranking mechanism to improve the search effectiveness.

Recommendation Systems (RS) (Jannach et al., 2010; Kantor et al., 2011; Adomavicius and Tuzhilin, 2005) have been a main focus of research, as these systems gradually reduce the existing information overload (information available on the Internet, data provided by devices/sensors of different types or other users, etc.), by recommending to the users personalized items of interest (e.g., movies, music, books, news, images, etc.) based on their preferences. However, in the field of RS, only a few works are marginally related to keyword-based searching. For example, two methods were studied for personalizing and improving the results of a social search engine (Shapira and Zabar, 2011), by using collaborative users' knowledge and integrating information from the user's social network; the proposed engine provides traditional keyword-based search functionalities. That work belongs to the field of IR, thus considering unstructured data sources, and applies recommendations to improve and customize the user experience. For movie recommendations, a hybrid system that alleviates the noise and semantic ambiguity problems present in keyword and tag representations of movies and user preferences was proposed (Stanescu et al., 2013). That proposal combines collaborative filtering and content-based recommendation techniques. As a final example, a study has been developed focused on improving the scalability and efficiency of a Big Data environment (Singam and Srinivasan, 2015). Specifically, the authors exploit keywords to indicate the preferences of users from a keyword candidate list, in order to generate appropriate recommendations based on a hybrid filtering algorithm.

As opposed to previous works, we specifically focus on the problem of correctly identifying the type of item required by a user when using a standard pull-based recommendation system. This first study in that direction represents a preliminary step forward for the development of complete and generic recommendation frameworks (del Carmen Rodríguez-Hernández and Ilarri, 2015).

3 HMM APPROACH

A Hidden Markov Model (Rabiner, 1989) can be defined as a triple A, B, π, where:

- $A = a_{ij}$ are the state transition probabilities.
- $B = [b_j(T)]$ are the observation probabilities (for each observation symbol T) at each state j.
- $\pi = [\pi_i]$ are the initial state probabilities.

There are mainly two basic problems associated to a Hidden Markov Model (Rabiner, 1989) which are relevant for the problem tackled in this paper:

1. Problem 1. Given the observation sequence $O = O_1, O_2, \ldots, O_T$, and the model λ, how do we choose a corresponding state sequence $Q = q_1, q_2, \ldots, q_T$ with the highest probability $P(Q|O,\lambda)$ (i.e., which best "explains" the observations)?
2. Problem 2. How do we adjust the model parameters $\lambda = (A, B, \pi)$ to maximize $P(O|\lambda)$?

According to the existing literature, the first problem (i.e., the problem of finding the most likely explanation for an observation sequence) can be solved efficiently using the Viterbi algorithm (Forney, 1973; Lou, 1995). To adapt that algorithm for our purposes, we have to define the following structures:

- Q is the set of states, which will be composed of the feature names (that characterize the items) and the item type.
- O is the set of observations, which will be the item types, as well the names and values of the item features.

As an example, we show in Figure 2 a fragment of the HMM proposed for a dataset InCarMusic (Baltrunas et al., 2011). The idea is the same for any other dataset.

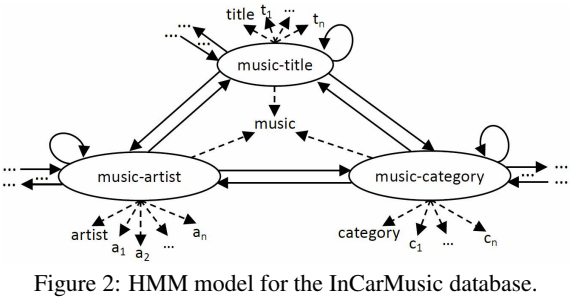


Figure 2: HMM model for the InCarMusic database.

From a reference database containing data on the domain ("item datasets" fragment in Figure 1), we extract a file "observations.txt" that contains the values and the names of the item features (e.g., title, artist and category in the case of Figure 2) and the item types (e.g., music in the case of Figure 2). Moreover, we extract a file "hmm_model.dat" with a specific structure. Considering the example of Figure 2, the structure of the file for three states (e.g., music-title, music-artist and music-category) and several observations (e.g., title, t_1, t_2, artist, a_1, a_2, a_3, category, c_1, c_2, music) is displayed in Figure 3.

In the model λ, each state contains the state transition probabilities A, the observation probabilities B,

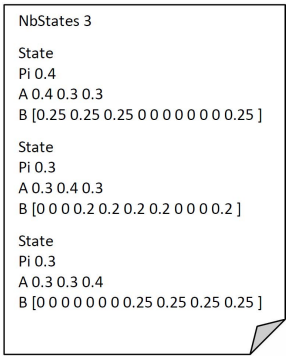


Figure 3: Example of a "hmm_model.dat" file structure.

and the initial state probabilities π. At the moment, by default, the probability values by state of the vector *B* are equally distributed on all the observations, dividing one by the number of terms related to the current state. Similarly, the state transition probabilities *A* have the same values for all the states, obtained by dividing one by the number of states. The initial state probabilities π are determined similarly. Nevertheless, our system supports the manual modification of the otherwise-equal values: the developer of a recommendation system can provide higher weights for certain elements that he/she considers more relevant, and the weights of the remaining elements will be readjusted to ensure that the sum of all the probabilities is still equal to one.

The keyword-based pull recommendation process proposed is presented in Figure 4, which summarizes the following sequence of steps:

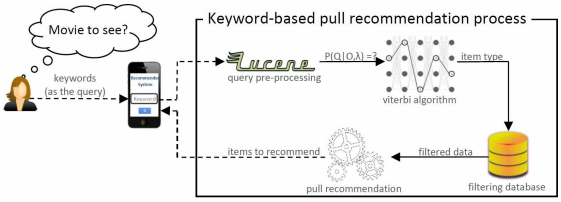


Figure 4: Keyword-based recommendations using HMM.

1. Input of the query: the user introduces the keywords as the input query in the Graphical User Interface (GUI).
2. Query pre-processing: the keywords are preprocessed by using the extension of an analyzer of Lucene, which applies the following filters:
   - Quotation tokenizer: it parses the query respecting the numbers and the double quotes.
   - Standard filter: it applies a standard tokenizer that parses the query into different types based on a grammar; for example, it splits words at punctuation characters, it removes punctuation, it splits words at hyphens (unless there is a number in the token), and it recognizes email addresses and internet host names as a single token.
   - Lower case filter: it normalizes the text of the token by converting it to lower case.
   - Stop filter: it removes stop words from the token streams, by using an input file containing stop words.
   - Snowball filter: it applies a filter that stems words using a Snowball-generated stemmer.
3. Application of the Viterbi algorithm: given the keywords as the observation sequence O and the HMM model λ, it allows determining the state sequence Q with the highest probability (e.g., music-title, music-artist, music-category, book_isbn, book_title, book_author, book_year, book_publisher, etc.).
4. Selection of the type of item: the type of item that the user needs would be determined by the highest-frequency state sequence (obtained in the previous step).
5. Filtering of the database: the database containing the different datasets is filtered by considering the type of item identified in the previous step (e.g., film, music, book, or concert). The data filtered will be used by the pull recommendation algorithm.
6. Application of the pull recommendation algorithm: it allows obtaining items of interest as an answer to the query submitted by the user, by applying any existing recommendation algorithm desired.
7. Display of the items recommended: a list of items recommended are provided to the user.

4 INFORMATION RETRIEVAL APPROACH

A second solution to consider to solve our general problem is the use of Information Retrieval (IR) techniques (Salton and McGill, 1986). In this case, the index of the retrieval engine contains a certain number of documents, whose content is automatically obtained from the databases that store the datasets (see Figure 1). Each document is named with the item type and the feature names. For example, for the dataset InCarMusic (Baltrunas et al., 2011), the document names to index are "music_title", "music_artist", and "music_category". The content of each document is composed of the values of the features (e.g., the artist

names, the music categories, and the music titles), the item type (e.g., music), and the names of the features (e.g., title, artist, and category). The structure of the documents to index is displayed in Figure 5.

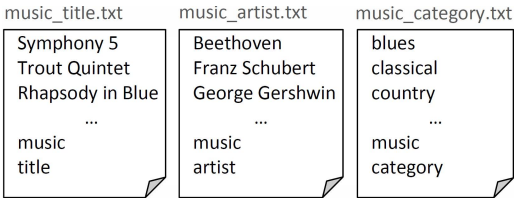


Figure 5: Example of the structure of the documents to index with the IR approach.

In general, the keyword-based pull recommendation process performs the following steps (see Figure 6):

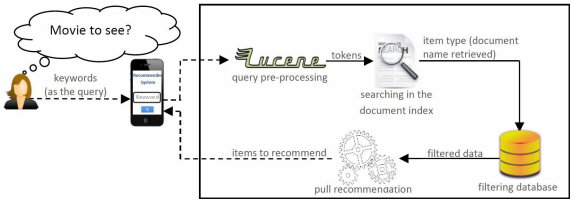


Figure 6: Keyword-based recommendations using IR.

1. Input of the query: the user introduces the keywords as the input query in the Graphical User Interface (GUI).
2. Query pre-processing: the keywords are preprocessed by using the same procedure described in Section 3.
3. Application of the Information Retrieval algorithm: given the input keywords, the system searches in the index the k documents that are most relevant to the query.
4. Selection of the type of item: the item type that the user needs would be the item type corresponding to the most relevant document (of the ranked list obtained).
5. Filtering of the database: the database containing the different datasets is filtered by considering the type of item identified in the previous step (e.g., film, music, book, or concert). The data filtered will be used by the pull recommendation algorithm.
6. Application of the pull recommendation algorithm: it allows obtaining items of interest as an answer to the query submitted by the user, by applying any existing recommendation algorithm desired.
7. Display of the items recommended: a list of items recommended are provided to the user.

5 EXPERIMENTAL EVALUATION

In this section, we present the experimental evaluation that we have performed to evaluate the two methods proposed. In Section 5.1, we describe the datasets that we use for the evaluation. In Section 5.2, we present the keyword-based queries that are evaluated. Then, we present the experimental settings and the evaluation results.

5.1 Datasets

We consider the following six datasets: LDOS-CoMoDa (Košir et al., 2011), InCarMusic (Baltrunas et al., 2011), Book-crossing (Ziegler et al., 2005), ConcertTweets (Adamopoulos and Tuzhilin, 2014), RCdata (Vargas-Govea et al., 2011) and Frappe (Baltrunas et al., 2015). In Table 1, some statistics related to the items of these datasets are described.

Table 1: Basic statistics of the datasets.

	LDOS-CoMoDa	InCarMusic	Book-crossing	ConcertTweets	RCdata	Frappe
Number of items	2513	139	271084	50971	130	4082
Number of attributes	8	7	7	8	25	10

In order to represent the HMM model, we used the item types, the feature names, and the feature values of the six datasets considered. However, we only chose the most appropriate features. Specifically, we ignored some features (name and values) of the following datasets: InCarMusic (e.g., album, mp3url, description, and imageurl), Book-crossing (e.g., image-URL-S, image-URL-M, and image-URL-L), ConcertTweets (e.g., URL), RCdata (e.g., fax, URL, and the-geom-meter), and Frappe (e.g., icon, description, and short description). We decided to ignore these features because they do not provide useful information. To experiment with datasets of equal size, we limit the number of instances considered from each dataset to 1000.

Considering the six datasets mentioned, the HMM model λ is composed of 52 states (e.g., film_director, music_artist, book_title, concert_date, restaurant_address, application_category, etc.). These states are the combination of the item types (e.g., film, music, book, concert, restaurant, application) and the feature names (e.g., director, artist, title, date, address, category) of the six datasets.

5.2 Queries

The two methods proposed (the one based on HMM and the one based on traditional IR) were evaluated by using 45 queries (see Table 2). Notice that some queries actually correspond to item types that are not available in the datasets considered. For those

queries, the best possible output is “other”, that represents a type of item not identified. For example, queries with identifiers from 30 to 35 explicitly include elements that are not part of the contents of the datasets.

Table 2: List of queries for evaluation.

Query Id	Original query
1	films similar to “toy story”
2	a romantic movie of the year 2009
3	videos of the director “walter lang”
4	the english film titled monkeys
5	a movie of the actor “diane ladd”
6	a music of the singer giovanni
7	rock song
8	music titled “für immer”
9	song of a rock artist
10	songs like “potato head blues” by “louis armstrong”
11	books about “seabiscuit”
12	books similar to ”fast women” by the author “jennifer crusie”
13	publications with an isbn number similar to 195153448
14	documents by the publisher scholastic
15	books with title “urban etiquette” and publisher ”wildcat canyon
16	concert of the band “iron maiden”
17	musical group that will play on ”02/04/2014” in Madrid
18	concerts in the venue “twickenham stadium”
19	the band direction in the state germany
20	concerts like “cattle decapitation” in ”cellular center”
21	publications with an isbn
22	place to eat
23	lodging in Modena
24	romantic melody
25	upcoming soccer matches in Barcelona
26	a recent horror movie
27	songs of movies
28	readings about movies
29	books about singers
30	self-help documents
31	romantic movie in 1949
32	policy documents of 1930
33	festivals in the region of the Holguin
34	movies that were premiered in 1927
35	documents of the Antarctica of 1908
36	restaurant with bar and permit smoking
37	place for dinner with an ambience familiar and low price
38	places opened in the hours of “12-00-22-00” to have lunch
39	restaurants with the name “taqueria el amigo”
40	french food and with “MasterCard Eurocard” payment
41	applications of photography
42	mobile applications developed by yahoo
43	chats similars to “whatsapp messenger”
44	“sport game” with many downloads
45	an apk similar to “Angry Birds” and language es

5.3 Implementation and Hardware

For the implementation of the HMM-based method, we used the Hidden Markov Model functionalities provided by the popular library *Apache Mahout* (http://mahout.apache.org/). Similarly, for the indexing of the documents for the IR-based method, we used *Apache Lucene* 2.4.0 (https://lucene.apache.org/). As explained in Sections 4 and 4, Lucene is also used for preprocessing (of input keywords and/or documents) in both methods.

Regarding the hardware, we used a standard standalone computer with the following features: Intel Core i5-2320 processor with 3 GHz and 16 GB of RAM, running Windows 7. We evaluated the performance of the proposals, although we omit the details due to space constraints. The latency is on the order of a few milliseconds (slightly higher for the IR approach) and the average memory consumption is around 9.5 MB (HMM) or 2.85 MB (IR).

5.4 Accuracy

The first proposed solution (based on HMM) computes the most likely state sequence matching an observation sequence given an HMM model. The second proposal (based on IR) searches the keywords in the query in the index of documents and returns a ranked list of hits. According to the values obtained of precision, recall and F1 measure in Tables 3 and 4, the HMM model performs better than the IR model in the experimental setup considered.

Nevertheless, it should be noticed that the IR approach is able to retrieve a ranked list of possible item types, but the HMM approach is only able to return one. Retrieving a top-K list could be interesting, as the user could quickly correct the item type identified as the most likely one if it is not correct. Although the Viterbi algorithm allows querying the probability of the most-likely sequence of states, it is not possible to retrieve the probability of all the possible sequences of states (which would be required in order to obtain a ranking).

Table 3: Evaluation of the HMM model.

Item type	Precision	Recall	F1
film	1.0	0.75	0.86
music	1.0	0.86	0.92
book	1.0	0.64	0.78
concert	0.71	0.83	0.77
restaurant	0.83	1.0	0.91
application	1.0	1.0	1.0
other	0.25	0.67	0.36
Average	0.83	0.82	0.80

Table 4: Evaluation of the IR model.

Item type	Precision	Recall	F1
film	0.67	0.75	0.71
music	1.0	0.43	0.6
book	0.78	0.64	0.7
concert	0.83	0.83	0.83
restaurant	0.71	1.0	0.71
application	0.56	1.0	0.86
other	0.0	0.0	0.0
Average	0.65	0.66	0.63

A possible problem with the HMM model is how to determine suitable probability values when the observation vector size is very large; potentially, it could happen that very small probabilities could be rounded

to zero if the global probability values are shared among many different possible values. Besides, better methods to assign the probabilities (by default we consider a proportional distribution/sharing) could be considered. Despite these concerns, the performance results obtained in the datasets evaluated so far are quite good.

5.5 Impact of the Number of Instances

We conducted another experiment with the aim of analyzing the impact of increasing the number of instances in the datasets on the performance of the two methods analyzed. For this experiment, we specifically selected the datasets Book-crossing, Concert-Tweets, and Frappe, which contain the larger number of instances (see Table 1). For each dataset, we considered four different versions (subsets of the original datasets) with an increasing number of instances (1000, 2000, 3000, and 4000 instances, respectively).

Then, we evaluated the performance (precision, recall, and F1-measure) of both models, as shown in Figures 7, 8, and 9. As shown in the figures, increasing the number of instances in the datasets in general leads to a decrease in the performance of both methods, but it is quite moderate and not very significant beyond 2000 instances per dataset. Again, in general the HMM-based method performs better than the IR-based approach.

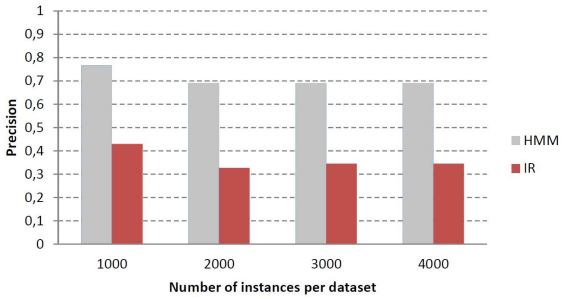


Figure 7: Average precision of both approaches.

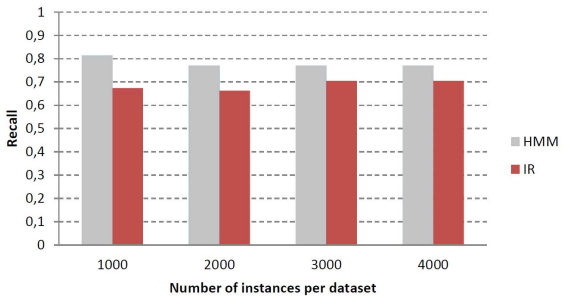


Figure 8: Average recall of both approaches.

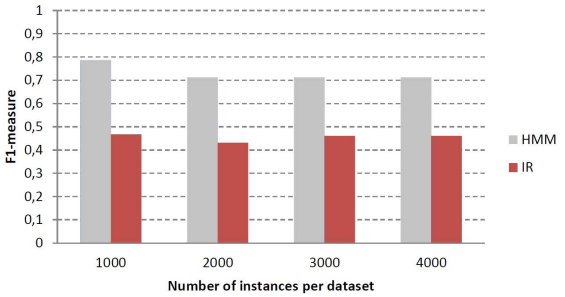


Figure 9: Average F1-measure of both approaches.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we have presented two methods for the identification of the type of item required by a user in a pull-based recommendation process. Up to the authors' knowledge, this is the first work that focuses on the problem of explicitly applying keyword-based techniques in a recommendation system scenario.

Despite the interest of the results obtained, this study is still preliminary and we can therefore envision several avenues of improvement. For example, a thesaurus can be used to obtain synonyms of keywords provided by the user. Similarly, key-phrases could be considered rather than only keywords. Besides, more experiments should be performed with a larger number of datasets (types of items) and a larger number of items. In particular, the problem that may arise if the sizes of the datasets are very different from each other should be analyzed, as a bias in favor of larger datasets may appear in the identification of the type of item.

ACKNOWLEDGEMENTS

This work has been supported by the CICYT project TIN2013-46238-C4-4-R, DGA-FSE, and the Keystone COST Action IC1302.

REFERENCES

Adamopoulos, P. and Tuzhilin, A. (2014). Estimating the value of multi-dimensional data sets in context-based recommender systems. In *Eighth ACM Conference on Recommender Systems (RecSys)*. CEUR.

Aditya, B., Bhalotia, G., Chakrabarti, S., Hulgeri, A., Nakhe, C., Parag, P., and Sudarshan, S. (2002). BANKS: Browsing and keyword searching in relational databases. In *28th International Conference*

on Very Large Data Bases (VLDB), pages 1083–1086. VLDB Endowment.

Adomavicius, G. and Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749.

Adomavicius, G. and Tuzhilin, A. (2011). Context-aware recommender systems. In Ricci, F., Rokach, L., Shapira, B., and Kantor, P. B., editors, *Recommender Systems Handbook*, pages 217–253. Springer.

Agrawal, S., Chaudhuri, S., and Das, G. (2002). DBXplorer: A system for keyword-based search over relational databases. In *18th International Conference on Data Engineering (ICDE)*, pages 5–16. IEEE.

Baltrunas, L., Church, K., Karatzoglou, A., and Oliver, N. (2015). Frappe: Understanding the usage and perception of mobile app recommendations in-the-wild. *CoRR*, abs/1505.03014.

Baltrunas, L., Kaminskas, M., Ludwig, B., Moling, O., Ricci, F., Aydin, A., Lüke, K.-H., and Schwaiger, R. (2011). InCarMusic: Context-aware music recommendations in a car. In *EC-Web*, volume 11, pages 89–100. Springer.

Bergamaschi, S., Domnori, E., Guerra, F., Orsini, M., Lado, R. T., and Velegrakis, Y. (2010). Keymantic: Semantic keyword-based searching in data integration systems. *Proceedings of the VLDB Endowment*, 3(1–2):1637–1640.

Bergamaschi, S., Guerra, F., Interlandi, M., Trillo-Lado, R., and Velegrakis, Y. (2013). QUEST: A keyword search system for relational data based on semantic and machine learning techniques. *Proceedings of the VLDB Endowment*, 6(12):1222–1225.

Bergamaschi, S., Guerra, F., Rota, S., and Velegrakis, Y. (2011). A Hidden Markov Model approach to keyword-based search over relational databases. In Jeusfeld, M., Delcambre, L., and Ling, T.-W., editors, *Conceptual Modeling–ER 2011*, volume 6998, pages 411–420. Springer.

Chakrabarti, S., Sarawagi, S., and Sudarshan, S. (2010). Enhancing search with structure. *IEEE Data Engineering Bulletin*, 33(1):3–24.

del Carmen Rodríguez-Hernández, M. and Ilarri, S. (2015). Pull-based recommendations in mobile environments. *Computer Standards & Interfaces*.

Forney, J. G. D. (1973). The Viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278.

Hristidis, V. and Papakonstantinou, Y. (2002). DISCOVER: Keyword search in relational databases. In *28th International Conference on Very Large Data Bases (VLDB)*, pages 670–681. VLDB Endowment.

Jannach, D., Zanker, M., Felfernig, A., and Friedrich, G. (2010). *Recommender Systems: An Introduction*. Cambridge University Press, first edition.

Kantor, P. B., Rokach, L., Ricci, F., and Shapira, B. (2011). *Recommender Systems Handbook*. Springer, New York, USA.

Košir, A., Odic, A., Kunaver, M., Tkalcic, M., and Tasic, J. F. (2011). Database for contextual personalization. *Elektrotehniški vestnik*, 78(5):270–274.

Levandoski, J. J., Sarwat, M., Eldawy, A., and Mokbel, M. F. (2012). LARS: A location-aware recommender system. In *28th International Conference on Data Engineering (ICDE)*, pages 450–461. IEEE.

Li, G., Ooi, B. C., Feng, J., Wang, J., and Zhou, L. (2008). EASE: An effective 3-in-1 keyword search method for unstructured, semi-structured and structured data. In *2008 ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pages 903–914. ACM.

Lou, H.-L. (1995). Implementing the Viterbi algorithm. *IEEE Signal Processing Magazine*, 12(5):42–52.

Rabiner, L. R. (1989). A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

Salton, G. and McGill, M. J. (1986). *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA.

Shapira, B. and Zabar, B. (2011). Personalized search: Integrating collaboration and social networks. *Journal of the American Society for Information Science and Technology*, 62(1):146–160.

Singam, J. A. and Srinivasan, S. (2015). Optimal keyword search for recommender system in Big Data application. *ARPN Journal of Engineering and Applied Sciences (ARPN-JEAS)*, 10(7):3243–3247.

Stanescu, A., Nagar, S., and Caragea, D. (2013). A hybrid recommender system: User profiling from keywords and ratings. In *2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, volume 1, pages 73–80. IEEE.

Teorey, T. J., Yang, D., and Fry, J. P. (1986). A logical design methodology for relational databases using the extended entity-relationship model. *ACM Computing Surveys*, 18(2):197–222.

Vargas-Govea, B., González-Serna, G., and Ponce-Medellın, R. (2011). Effects of relevant contextual features in the performance of a restaurant recommender system. In *Fifth ACM Conference on Recommender Systems (RecSys): Third Workshop on Context-Aware Recommender Systems (CARS)*, volume 791. CEUR.

Ziegler, C.-N., McNee, S. M., Konstan, J. A., and Lausen, G. (2005). Improving recommendation lists through topic diversification. In *14th International Conference on World Wide Web (WWW)*, pages 22–32. ACM.

Zobel, J. and Moffat, A. (2006). Inverted files for text search engines. *ACM Computing Surveys (CSUR)*, 38(2):6.

On the Design of a Traffic Observatory Application based on Bus Trajectories

Kathrin Rodriguez[1], Marco A. Casanova[1], Luiz André Paes Leme[2], Hélio Lopes[1], Rafael Nasser[1] and Bruno Guberfain do Amaral[1]

[1]*Department of Informatics – Pontifical Catholic University of Rio de Janeiro, RJ, Brazil*

[2]*Institute of Computing, Fluminense Federal University, Niteroi, RJ, Brazil*

{kllanes, casanova}@inf.puc-rio.br, lapaesleme@ic.uff.br, lopes@inf.puc-rio.br, rafael.nasser@les.inf.puc-rio.br, bguberfain@inf.puc-rio.br

Keywords: Traffic Modelling, Trajectory Data Mining, Data Stream Processing.

Abstract: Buses, equipped with active GPS devices that continuously transmit their positions, can be understood as mobile traffic sensors. Indeed, bus trajectories provide a useful data source for analyzing traffic, if the city is served by a dense bus network and the city traffic authority makes the bus trajectories available openly, timely and in a continuous way. This paper explores the design of a traffic observatory application based on bus trajectories, defined as an application developed to detect when the traffic patterns of selected streets of a city, observed during certain periods of time, deviate from the typical traffic patterns. The major contributions of the paper are a list of requirements for traffic observatory applications, a detailed discussion of key operations on bus trajectories and a description of experiments with a traffic observatory prototype using bus trajectories made available by the traffic authority of the City of Rio de Janeiro.

1 INTRODUCTION

An intelligent control and management system, that has a data-driven approach for modelling, analysis, and decision-making (Zhang et al. 2013), may help achieve better traffic control and create mobility plans. As their main input data, such systems adopt trajectories, generated by GPS devices installed in vehicles (Shi et al. 2008), such as taxis (Zhu and Xu 2015) and buses (Sunil et al. 2014).

Indeed, buses, equipped with active GPS devices that continuously transmit their position, can be understood as mobile traffic sensors. A *raw bus trajectory* is a continuous data stream acquired from such a GPS device.

Bus trajectories provide a useful data source for analyzing traffic, if the city is served by a dense bus network and the city traffic authority makes the bus trajectories available openly, timely and in a continuous way. Under such conditions, bus trajectories are a better data source to analyze traffic than data generated by proprietary traffic applications that acquire the position of private cars and that depend on drivers' volunteered traffic feedback. Indeed, bus trajectories are a stable data source, in the sense that they cover the same set of streets, at predictable regular intervals, if traffic conditions permit. In fact, this is the point: if the buses in a given area are not running according to the usual schedule, then a traffic perturbation is the most probable cause. Furthermore, if stored in an adequate way, bus trajectories will provide, over time, a historical picture of how the city evolved, much in the same way as satellite imagery gives a historical picture of how an urban area grew.

This paper explores the design of a *traffic observatory application* based on bus trajectories, defined as an application developed to detect when the traffic patterns of selected streets of a city, observed during certain periods of time, deviate from the typical traffic patterns. The design of such application poses at least the following challenges: (1) How to analyze the bus network (served by GPS-equipped buses) to select streets whose traffic can be monitored with the help of the bus trajectories; (2) How to mine a bus trajectory dataset to uncover traffic patterns; (3) How to detect traffic anomalies, estimate their impact and provide explanations, using data sources other than the bus trajectories; (4) How to maintain and compare different versions of the street network, the bus network and the traffic patterns, to help city planners assess changes.

Rodriguez, K., Casanova, M., Leme, L., Lopes, H., Nasser, R. and Amaral, B.
On the Design of a Traffic Observatory Application based on Bus Trajectories.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 215-222
ISBN: 978-989-758-187-8

The major contributions of this paper are three-fold: (1) a list of requirements for traffic observatory applications; (2) a detailed discussion of key operations on bus trajectories; (3) a description of experiments with a traffic observatory prototype using raw bus trajectories made available by the traffic authority of the City of Rio de Janeiro, which corroborate the usefulness of the proposed approach to monitor traffic.

The rest of this paper is organized as follows. Section 2 lists the basic requirements for a traffic observatory application. Section 3 introduces the main concepts related to street networks, bus networks and trajectories. Section 4 discusses some key operations on bus trajectories. Section 5 describes experiments with real data. Section 6 covers related work. Finally, Section 7 contains the conclusions.

2 REQUIREMENTS FOR A TRAFFIC OBSERVATORY APPLICATION

In this section, we enumerate the major requirements that a traffic observatory application, based on bus trajectories, must meet.

(1) Select Streets Whose Traffic can be Monitored with the Help of Bus Trajectories.

The first requirement quite simply reflects the fact that the traffic sensors are the buses equipped with GPS. The application must be able to analyze the bus network and select those streets that are frequently crossed by buses and whose traffic can, therefore, be monitored by analyzing the bus trajectories.

This requirement depends on a clear definition of the concepts of *street network, bus network* and *monitored street network*, the subset of the street network that can monitored by analyzing the bus trajectories.

(2) Discover Traffic Patterns.

The second requirement refers to the basic question of defining what should be considered normal versus abnormal traffic behavior. Thus, the application must be able to mine a bus trajectory dataset to discover traffic patterns for select street segments, over a given period of time.

In addition to the definition of street network and bus network, this requirement depends on the concepts of *traffic patterns*, in the form of *traffic flow patterns* and *travel time patterns*.

(3) Detect and Explain Traffic Anomalies, and Estimate their Impact.

The third requirement covers the core of a traffic observatory application. It refers to monitoring the traffic for select street segments, over a given period of time, to uncover the observed traffic patterns, compare them with typical traffic patterns and, finally, mark the observed traffic patterns that deviate from the typical patterns above a given threshold.

Furthermore, this requirement includes identifying traffic events from additional data sources that might cause the deviations.

(4) Maintain and Compare Different Versions of the Street Network, the bus Network and the Traffic Patterns.

The last requirement imposes that the application must maintain versions of the street network, the bus network, the monitored street network and the traffic patterns. The application must also support comparing different versions of the street and bus networks to assess the impact of changes on select street segments, which provide a useful tool for city planners.

3 STREET NETWORKS AND TRAJECTORIES

In this section, we define the concepts identified in Section 2 and introduce the concept of trajectory.

3.1 Street Networks

For the purposes of this paper, a *geo-referenced street network* is modelled as a labelled, directed graph $G=(V,E,nl,el)$, where the node labelling function *nl* associates a geo-referenced point (in an appropriate geographic coordinate system) with each node in V and the edge labelling function *el* assigns a geo-referenced line segment (in the same the geographic coordinate system used by the node labelling function) to each edge in E. Intuitively, the edges represent street segments and the nodes indicate the start and end points of the street segments; these labelling functions must therefore be consistent with each-other. A street network may have other labelling functions, such as the street name to which the segment belongs.

The familiar notions of (*directed*) *path* and *circuit* from graph theory directly apply to street networks. A *street route* is simply a path in G.

A *flow pattern* for a node n in V is pair $\varphi = (\delta,\pi)$, where δ is a representation of the distribution of the flow of vehicles that pass by n and π is a specification

of the domain of δ (to facilitate comparing patterns). Likewise, a *travel time pattern* for a path p of G is a pair $\tau = (\gamma,\pi)$, where γ is a representation of the time distribution of the travel time of vehicles that traverse p and π is a specification of the domain of γ. Figure 2 at the end of the paper presents an example of travel time patterns.

Given a street network $G=(V,E,nl,el)$, a *bus line* of G is a set l of paths in G; each path in l is called a *bus route of l*. That is, a bus line may have several alternative routes, which depend on the time of the day, for example. A *geo-referenced bus stop*, or simply a *bus stop,* of a bus route r is a point of a line segment that labels an edge in r. Given a street network $G=(V,E,nl,el)$, a *bus network of G* is a pair $N=(L,S)$, where L is a set of bus lines of G and S is a set of bus stops of the bus routes of the bus lines in L.

Finally, a *monitored street network* for a street network $G=(V,E,nl,el)$ and a bus network $N=(L,S)$ of G is a quadruple $M=(W,F,wl,fl)$ such that:

- $W \subseteq V$
- If (m,n) is in F then there is a bus route of a bus line in N that connects m and n
- wl is a function that labels each node n in W with the bus routes that pass through n
- fl is a function that labels each edge $f=(m,n)$ in F with the bus routes that that connects m and n

3.2 Trajectories

A *trajectory* is the representation of the position evolution of a moving object. We can have different representation levels: at the *raw trajectory level*, the sequence of sample points is represented as collected by the mobile device whereas, at the *segmented trajectory level*, homogeneous parts of a raw trajectory are identified based on some criterion.

More precisely, a *geo-referenced spatio-temporal point* is a pair $((x,y),t)$, where (x,y) is a geo-referenced point and t is a timestamp. A *raw trajectory* of a moving object is a sequence of geo-referenced spatio-temporal points, $s=((p_1,t_1),(p_2,t_2),...,(p_n,t_n))$, such that t_i is less than t_{i+1}, for $i=1,...,n-1$. A *segment c* of a raw trajectory s is a subsequence of s. Finally, a *segmented trajectory* of a raw trajectory s is sequence $g=(g_1,...,g_h)$ of segments of s such that s is the concatenation of $g_1,...,g_h$.

Since a bus is a moving object b, a *raw bus trajectory s* of b is simply a raw trajectory generated by b. Useful strategies to segment s would be based on the bus stops of a route of the bus line, the nodes of the monitored street network, or other control points. The next section discusses this last segmentation criterion in detail.

4 SOME KEY OPERATIONS OF A TRAFFIC OBSERVATORY

This section briefly discusses the following operations: segmentation of raw bus trajectories; detection of travel time anomalies; estimation of travel time delays; and finding explanations for travel time anomalies. These operations are at the heart of the traffic observatory prototype illustrated in Section 5. Other equally important operations, such as mining traffic patterns, will not be covered due to space limitations.

4.1 Segmentation of Raw Bus Trajectories

The *real-time control points segmentation problem* is defined as follows:

- Let R be a bus route and $n_1,\ldots,n_k$ be a list of *control points* that succeed each other along R. Given a raw trajectory s, generated by a bus b which follows bus route R, segment s into $g=(g_1,...,g_{k-1})$, in real-time, so that g_i corresponds to the segment of s that starts in a point q_i closest to n_i and ends in a point q_{i+1} closest to n_{i+1}, for i=1,…,k-1.

The control points may be arbitrarily chosen along the bus route R, they may be the bus stops of R or they may correspond to points pre-defined in a monitored street network. However, we assume that n_i immediately precedes n_{i+1} in R, for i=1,…,k-1.

By segmenting *s in real-time* we mean that the spatial-temporal points of s are processed as a data stream, that is, at time t, the segmentation algorithm has access only to the prefix of s up to t.

There are several practical problems to take into account, such as:

(1) The bus route associated with s may be incorrect.
(2) GPS devices introduce errors.
(3) The sampling interval at which the GPS points are acquired may be too long.

We assume that Problems (1) and (2) have been solved by a pre-processing step so that the bus route R is correct and all points in s fall over R.

Problem (3) deserves a separate discussion. If the sampling interval at which the GPS points are acquired is too long, or the bus is running too fast, no point in the trajectory s may correspond exactly to any of the control points. Given a control point n_i, there are at least three possible solutions: (1) select the last point q_i in s that occurs before n_i along R; (2) select the first point r_i in s after n_i; (3) use the timestamps of q_i and r_i to generate a timestamp u_i by interpolation

and artificially add (n_i, u_i) to s. Any of these solutions actually use route R to impose a linear order on the trajectory points together with the control points.

In the rest of this section, we briefly discuss a real-time control points segmentation strategy based on the first option, for the sake of simplicity.

Let $s = ((p_1, t_1), (p_2, t_2), ..., (p_n, t_n))$ be a raw bus trajectory generated by a bus that follows bus route R. Assume that the points in s correctly fall over R.

Suppose that we have already processed the prefix $((p_1, t_1), (p_2, t_2), ..., (p_i, t_i))$ of s and that (p_i, t_i) is such that p_i is the last point in s before n_i. We must discover (p_j, t_j) in s such that p_j is the last (spatial) point before n_{i+1} along R. We will then have found the desired segment g_i, which is $((p_i, t_i), ..., (p_j, t_j))$.

To discover one such point, we associate with (n_i, n_{i+1}) a variable C, which is initially *Null*, and which will hold a pair (p_h, t_h), where (p_h, t_h) is the last known point of s.

Let (p_k, t_k) be a new spatial-temporal point of s, that is, (p_k, t_k) is added at the end of the current prefix of s. There are two cases to consider:

1. p_k lies before n_{i+1} along R. Then, update C to (p_k, t_k).
2. p_k lies after n_{i+1} along R. Then, (p_k, t_k) is the first point in s after n_{i+1} and the current value of C is used as the end-point of the segment that started on (p_i, t_i).

A few comments are worth at this point. As already indicated, this segmentation strategy depends on a pre-processing step so that the bus route R that is correct and all points in s fall over R.

The real-time control points segmentation strategy can be modified to simultaneously segment a set of raw trajectories that traverse the same control points (see examples in Section 5.3) simply by replacing variable C by a hash table whose key is the bus ID.

Also, the strategy can be used to (off-line) segment a set of raw trajectories stored in a trajectory dataset. Furthermore, with minor modifications, the segmentation strategy can be transformed into a strategy to monitor buses whose routes cover a given set of control points.

4.2 Detecting Travel Time Anomalies

The *real-time travel time anomaly detection problem* is defined as follows:

- Given a street route S and a time interval T, detect in real-time if the travel time to traverse S during T is deviating from the average travel time.

An example of a time interval T would be "Monday, August 17th, from 6:00 AM to 10:00 AM". We also say that a time interval U, such as "Monday, August 10th from 6:00 AM to 10:00 AM", is *consistent with T*.

We recall that both a street route S and a bus route R are paths of the street network. We say that a bus route R *matches* S iff S is a sub-path of R (this notion is needed to select bus trajectories that cross S).

Let S be a street route and assume that S starts on a node labelled with point n_i and ends on a node labelled with point n_{i+1}. Let T be a time interval. Let π be a set of trajectories that are being generated, during the time interval T, by buses that follow routes that match S.

A real-time traffic anomaly detection strategy, similar to the segmentation strategy described in Section 4.1, would go as follows:

1. Off-line, as a preparation step, obtain an estimation for the average travel time to traverse S, denoted $\bar{\tau}[S, \alpha, T, P]$, using the travel times to traverse S observed in a set α of archived trajectories, for time intervals consistent with T, over a period of time P.
2. In real-time, given a trajectory s in π, suppose that the prefix $((p_1, t_1), (p_2, t_2), ..., (p_i, t_i))$ of s has already been processed and that (p_i, t_i) is such that p_i is the spatial point in s closest to n_i. When a point (p_k, t_k) of s is received, if $t_k - t_i > \bar{\tau}[S, \alpha, T, P]$, then the bus that is generating s is running late to reach n_{i+1}, that is, to traverse S.
3. If more than one bus, but less than Y buses are running late to traverse S, raise a *yellow semaphore* where Y is a given constant.
4. If more than Y buses are running late, raise a *red semaphore*.

The use of semaphores is justified since buses might be delayed for a number of reasons and, hence, one cannot signal that there is a travel time anomaly to traverse S at T if just one bus is running late.

4.3 Estimating Travel Time Delays

The *travel time delay estimation problem* is defined as follows:

- Given a street route S and two periods of time P_1 and P_2, estimate the differences between the travel times to traverse S at P_1 and at P_2.

A quite simple travel time delay estimation strategy would go as follows:

1. Select a set α_k of trajectories from a set of archived trajectories such that the trajectories match S and cover P_k, for k=1,2.

2. Obtain an estimation for the distribution of travel times to traverse S, denoted $\tau_k[S, \alpha_k, P_k]$, using the travel times to traverse S at P_k observed in the trajectories in α_k, for k=1,2.
3. Compare $\tau_1[S, \alpha_1, P_1]$ and $\tau_2[S, \alpha_2, P_2]$.

Section 5.3 provides examples of travel time delay estimations.

Finally, using travel time delay estimations, it would also be possible to estimate the number of bus passengers affected, or the total loss of time (incurred by bus passengers), if bus passenger data were available.

4.4 Finding Explanations for Travel Time Anomalies

The *explanation of travel time anomalies* problem is defined as follows:

- Given a street route S and a time interval T such that a travel time anomaly has been detected, find a traffic event that can explain the anomaly.

A strategy to address this problem involves interpreting tweets that describe traffic-related events and that are distributed by government agencies or by news agencies (blind1). Briefly, the strategy would go as follows:

1. Suppose that a travel time anomaly has been detected for a given street route S and a time interval T.
2. Use the labelling functions of the street network to find the street names of the edges that compose the street route S.
3. Search the appropriate Twitter channels to find tweets that refer to traffic events that occurred in such streets during T; the search requires interpreting the tweets to identify street names and other traffic event details (blind1).
4. If no such tweets are found, use the street network to find the neighboring streets along street route S, up to a certain distance, and repeat Step (3).
5. Output any tweet found.

Section 5.3 provide an example of a traffic event that caused a considerable traffic time anomaly.

5 EXPERIMENTS

This section describes experiments with the traffic observatory prototype developed to test the concepts introduced in previous sections.

5.1 The Bus Network of the City of Rio De Janeiro, Brazil

The public transportation system of the City of Rio de Janeiro is largely based on buses. The statistics published for the year 2014 are the following:

- Bus lines: 716
- Number of buses: 8,916
- Number of trips:18.5 million
- Number of passengers transported: 1,263 million
- Kilometers travelled: 760 million
- Number of companies: 44
- Number of employees: 41,375
- Average bus age: 4.06 years
- Average no. of passengers per kilometer: 1.39
- Average no. of kilometers travelled per bus per month: 7,094

Yet more expressive is the fact that buses accounted for nearly 60% of all passengers transported over the past three years.

5.2 Data Collection and Visualization

The traffic observatory prototype offers a basic data collection service that:

1. Captures the bus lines, bus routes and bus stops from the traffic authority Web site.
2. Captures, at regular interval, the raw bus GPS points from the traffic authority Web site.
3. Keeps in core the last 5 positions of each bus.
4. Stores in secondary storage all points captured, organized by day.

From June 12th, 2014 until December 1st, 2015, the service collected more than 2 billion records.

The traffic observatory prototype also offers simple visualization services that allow users to overlay bus trajectory data on top of a street map of the city:

1. The last known position of each (operational) bus.
2. The last known position of each bus, up to a 10-minute delay.
3. The last known position of each bus of a given bus line, together with the actual bus route (forward and return).
4. The last 5 positions of a specific bus, together with the actual bus route (forward and return).

In all cases, the user may obtain the data associated with a bus by passing the mouse over the icon that represents the bus.

5.3 An Example of Travel Time Delay Estimation

To illustrate what one can expect from the traffic observatory prototype, we estimate the travel time delays caused by a traffic accident that occurred in the metropolitan area of the City of Rio de Janeiro.

The accident was a fatal collision that caused the death of a motorcyclist at the Zuzu Angel Tunnel, which is part of an expressway that connects the south and the west zones of Rio. As shown in Figure 1, the accident occurred on Monday, August 17th, 2015 and took place at, approximately, latitude -22.992342 and longitude -43.249278 (near the Rocinha community in the São Conrado area).

To evaluate the impact of this event in term of travel time delays, the road segments analyzed were: Zuzu Angel Tunnel, Jardim Botânico Street and Bartolomeu Mitre Avenue, identified in Figure 1 in blue, green and red, respectively. These segments were chosen based on the (crucial) nodes, shown in Figure 1, of the monitored street network of Rio de Janeiro.

Figure 2 shows the travel time spent to traverse the Zuzu Angel Tunnel on the day of the accident versus the typical travel time pattern for the segment, mined from the archived bus trajectories, for the same day of the week (i.e., Mondays). As the graph in this figure reveals, this event caused considerable travel time delays for a crucial period of the day. The travel time delays reached a peek of nearly 30 minutes at 8:00 AM and were observed for nearly four hours, from 6:00 AM to 10:00 AM. Travel time delays were also observed throughout the Jardim Botânico Street up to the Rebouças Tunnel (indicated by top most dot in Figure 1), located 10 km from the accident site.

To conclude, this example illustrates the ability of the traffic observatory prototype to mine a trajectory dataset to uncover typical and abnormal traffic patterns for selected road segments and time periods and to compare the patterns to assess travel time delays (Figure 2 shows typical patterns in green, or light grey, and abnormal patterns in red, or dark grey).

6 RELATED WORK

The segmentation of raw trajectories may use different criteria, ranging from the transportation means used (Biljecki et al., 2013; Biljecki, 2010), potential-transition locations (e.g. bus stops) (Liao, 2006), geo-spatiotemporal information (Buchin et al., 2015; Yoon et al., 2008), detection of similar sub-trajectories (Sankararaman et al., 2013) and movement analysis (Alewijnse et al., 2014; Buchin et al., 2012). Section 4.1 specifically discussed how to segment row trajectories based on the passing of buses by control points.

Estimating traffic patterns from GPS data streams is an important task to improve the efficiency of traffic systems. According to (Zhang et al., 2013), traffic applications using GPS data streams can be divided into two main groups: centralized and distributed. The first group uses traffic data from multiple GPS devices simultaneously, while the second group of applications uses individual GPS data. Traffic state estimation (Geisler et al., 2012), queue profile estimation (Ramezani and Geroliminis, 2015), detection of traffic anomalies (Kuang et al., 2015) are examples of applications of the centralized applications. Applications of the second group include: vehicle performance analysis (Kargupta et al., 2010), vehicle monitoring (Jose et al., 2015), and vehicle anomaly detection (Chen et al., 2012). This paper could be classified in the first group of applications, as it analyses traffic based on multiple GPS-enabled vehicles.

Kumar et al. (2005) presented a real-time surveillance system with a rule-based behavior and event-recognition module for traffic videos. Lu et al. (2008) developed HOLMES, which is a system for highway operation monitoring and evaluation.

Concerning bus transportation, several works addressed the problem of determining the estimated time of arrival (Bullock, Jiang and Stopher, 2005; Sun et al., 2007). Kormaksson et al. (2014) presented a specific study about the City of Rio de Janeiro.

7 CONCLUSIONS

We argued that buses, equipped with active GPS devices that continuously transmit their position, can be understood as mobile traffic sensors. Indeed, bus trajectories provide a useful data source for analyzing traffic, if the city is served by a dense bus network and the city traffic authority makes the bus trajectories available openly, timely and in a continuous way.

We briefly listed the fundamental requirements for traffic observatory applications. Then, we discussed some key operations on bus trajectories. Finally, we described experiments with a traffic observatory prototype using bus trajectories made available by the traffic authority of the City of Rio de Janeiro. The results obtained corroborate the usefulness of using bus trajectories to monitor traffic.

As for future work, we are gradually increasing the functionality of the traffic observatory prototype to cover all requirements listed in Section 2.

ACKNOWLEDGEMENTS

This work was partly funded by CNPq under grants 153908/2015-7, 557128/2009-9, 444976/2014-0, 303332/2013-1, 442338/2014-7 and 248743/2013-9 and by FAPERJ under grants e E-26-170028/2008 and E-26/201.337/2014.

REFERENCES

Albuquerque, F.C., Casanova, M.A., Lopes, H., Redlich, L.R., Macedo, J.A.F., Lemos, M., Carvalho, M.T.M., Renso, C. A methodology for traffic-related Twitter messages interpretation. Computers in Industry. doi: 10.1016/j.compind.2015.10.005.

Alewijnse, S., Buchin, K., Buchin, M., Kolzsch, A., Kruckenberg, H. and Westenberg, M. 2014. A framework for trajectory segmentation by stable criteria. Proc. 22nd ACM SIGSPATIAL Int'l. Conf. on Advances in Geographic Information Systems, 351–360.

Biljecki, F., Ledoux, H. and Van Oosterom, P. 2013. Transportation mode-based segmentation and classification of movement trajectories. *Int'l. J. of Geographical Information Science*, Vol. 27, No. 2, 385–407.

Biljecki, F. 2010. Automatic segmentation and classification of movement trajectories for transportation modes. TU Delft, Delft University of Technology.

Buchin, M., Driemel, A., van Kreveld, M. and Sacristan, V. 2015. Segmenting trajectories: A framework and algorithms using spatiotemporal criteria. *J. Spatial Information Science*, No. 3, 33–63.

Buchin, M., Kruckenberg, H. and Kolzsch, A. 2012. Segmenting trajectories based on movement states. Proc. 15th Int'l. Symp. Spatial Data Handling (SDH), 15–25.

Bullock, P., Jiang, Q., Stopher, P. R. 2005. Using GPS technology to measure on-time running of scheduled bus services. *Journal of Public Transportation*, Vol. 8, No. 1, p. 21–40.

Chen, C., Zhang, D., Castro, P. S., Li, N., Sun, L., and Li, S. 2012. Real-time detection of anomalous taxi trajectories from GPS traces. *Mobile and Ubiquitous Systems: Computing, Networking, and Services*, 63–74.

Geisler, S., Quix, C., Schiffer, S., and Jarke, M. 2012. An evaluation framework for traffic information systems based on data streams. *Transportation Research Part C: Emerging Technologies*, Vol. 23, 29–55.

Sunil, N., Ajinkya, V., Swapnil, C., and Vyankatesh, B. 2014. Dynamic bus timetable using GPS. *Int'l. J. of Advanced Research in Computer Engineering & Technology*, Vol. 3, No. 3.

Jose, D., Prasad, S., and Sridhar, V. 2015. Intelligent vehicle monitoring using global positioning system and cloud computing. *Procedia Computer Science*, Vol. 50, 440–446.

Kargupta, H., Sarkar, K., and Gilligan, M. 2010. Minefleet R : An overview of a widely adopted distributed vehicle performance data mining system. Proc. 16th ACM SIGKDD Int'l. Conf. Knowledge Discovery and Data Mining, 37–46.

Kormaksson, M. et al. 2014. Bus Travel Time Predictions Using Additive Models. Proc. 2014 IEEE Int'l. Conf. on Data Mining (ICDM), 875–880.

Kumar, P. et al. 2005. Framework for real-time behavior interpretation from traffic video. *IEEE Trans. On Intelligent Transportation Systems*, Vol. 6, No. 1, 43–53.

Kuang, W., An, S., and Jiang, H. 2015. Detecting traffic anomalies in urban areas using taxi GPS data. *Mathematical Problems in Engineering*, 501:809582.

Liao, L., Patterson, D.J., Fox, D. and Kautz, H. 2006. Building personal maps from GPS data. *Annals of the New York Academy of Sciences*, Vol. 1093, No. 1, 249–265.

Lu, C.-T. et al. 2008. Homes: highway operation monitoring and evaluation system. Proc. 16th ACM SIGSPATIAL Int'l. Conf. on Advances in Geographic Information Systems., 85.

Ramezani, M. and Geroliminis, N. 2015. Queue profile estimation in congested urban networks with probe data. *Computer-Aided Civil and Infrastructure Engineering*, Vol. 30, No. 6, 414–432.

Sankararaman, S., Agarwal, P. K., Molhave, T., Pan, J. and Boedihardjo, A.P. 2013. Model-driven matching and segmentation of trajectories, Proc. 21st ACM SIGSPATIAL Int'l. Conf. on Advances in Geographic Information Systems, 234–243.

Shi, W., Kong, Q.-J., and Liu, Y. 2008. A GPS/GIS integrated system for urban traffic flow analysis. Proc. 11th Int'l. IEEE Conf. on Intelligent Transportation Systems, 844–849.

Sun, D. et al. 2007. Predicting Bus Arrival Time on the Basis of Global Positioning System Data. *Transportation Research Record*, Vol. 2034, No. 1, 62–72.

Yoon, H. and Shahabi, C. 2008. Robust time-referenced segmentation of moving object trajectories. Proc. 8th IEEE Int'l. Conf. on Data Mining, 1121–1126.

Zhang, J.-D., Xu, J., and Liao, S. S. 2013. Aggregating and sampling methods for processing GPS data streams for traffic state estimation. *IEEE Trans. on Intelligent Transportation Systems*, Vol. 14, No. 4, 1629–1641.

Zhu, B. and Xu, X. 2015. Urban principal traffic flow analysis based on taxi trajectories mining. *Advances in Swarm and Computational Intelligence*, 172–181.

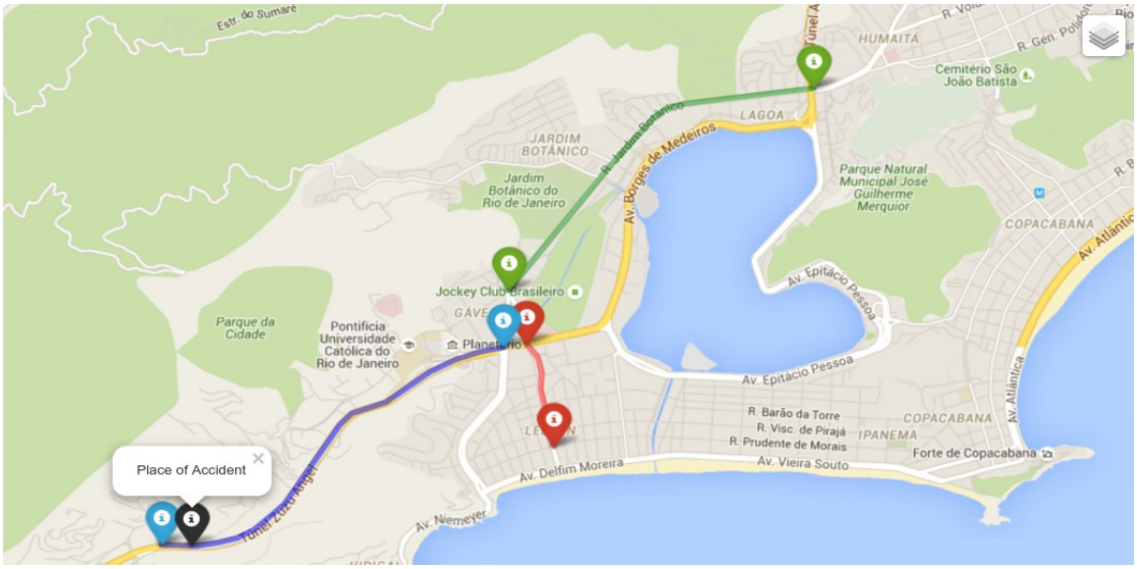


Figure 1: Place of the accident.

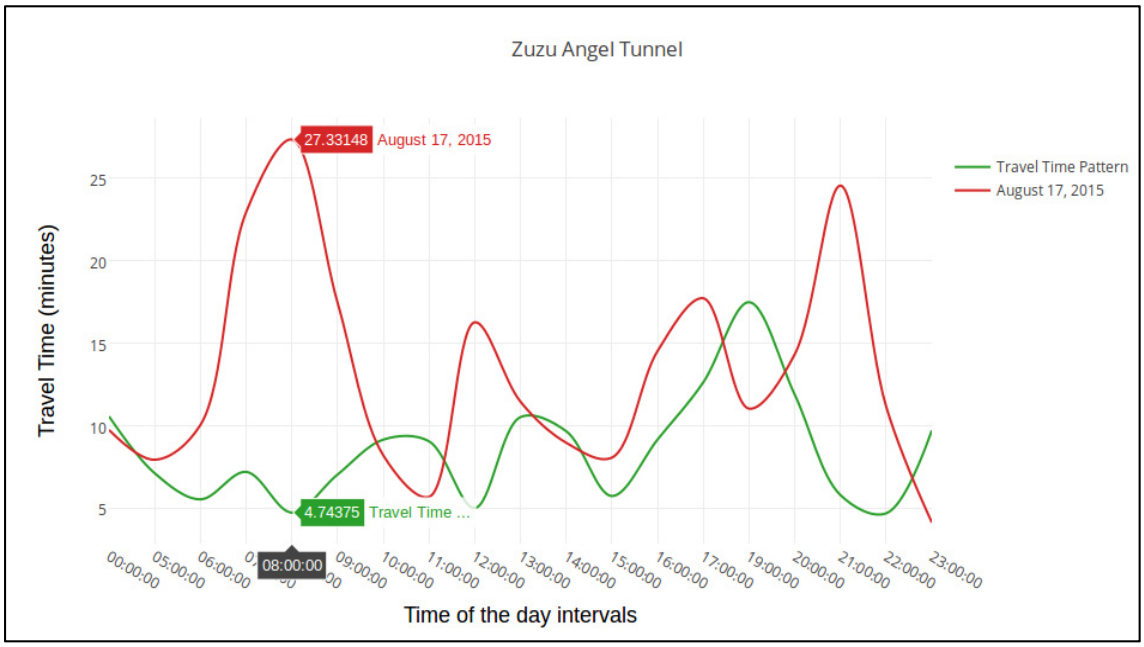


Figure 2: Travel Time Pattern vs Travel Time at the day of accident – Zuzu Angel Tunnel.

Adaptation Services-oriented Systems LifeCycle

I. Elmagrouni[1], A. Kenzi[2], M. Lethrech[1] and A. Kriouile[1]
[1]*SIME Laboratory, ENSIAS, Mohammed V University, Rabat, Morocco*
[2]*Sidi Mohamed Ben Abdellah University, Fes, Morocco*
{issam.elmagrouni, adil.kenzi}@gmail.com, mohammed.lethrech@um5s.net.ma, kriouile@ensias.ma

Keywords: Development Process, SOC, CAC, SOA / Web Services, MDA.

Abstract: This work presents the approach of the Development of adaptable Services-oriented Systems. Not the only adaptability is important for the survival and success of Services-oriented Systems, but it is also important for the rapid changes in technology, organizational structure, human perception and needs. This paper discusses the shortcomings of current solutions for adaptive service-oriented System. To address those shortcomings, some techniques are used to build and evolve proactive Services-oriented Systems. Using those techniques in an integrated way is described along the phases of the service lifecycle.

1 INTRODUCTION

The service-oriented computing (SOC) uses services as basic constructs to support the rapid development of low cost, and easy composition of distributed applications even in heterogeneouscomputing environments.SOCperspectiveis to join services in a network of loosely coupled services, create flexible business processes and agile applications that can span different organizations and computing platforms.Besides, CAC (Context-Aware Computing) has been proposed to adapt applications and software systems to different useful contexts. These cover all the information that characterizes the situation of an entity. An entity can be a person, place, or object that may be relevant to the interaction between the user and the application. One of the challenges oflifecycleservicesis the stage that identifies the services that support the business activities of the organization.There are many objectives of this paper. Firstly,is the definition of a development process to guide the development of adaptable SOS from business requirements. Secondly, is the definition of phases, activities, and artifacts that allow the identification, specification and implementation of adaptive services.The life-cycle models; for Service-Based applications that have been presented in the literature (examples include SLDC, RUP for SOA, SOMA, and SOAD) are mainly focused on the phases that precede the release of software; and even in the cases in which they focus on the operation phases, they usually do not consider the possibility for to adapt dynamically to new situations, contexts, requirement needs, service faults, etc.Specifically, the following aspects have not been yet considered in those life-cycle models: Requirements elicitation, design for adaptation.

The first section briefly presents the work and the approaches that are related to the study. The second section describes the case study to illustrate the approach. The Third section focuses on the process definition which will enable the development of adaptive services. Last but not least, the article will be summed up with a conclusion and outlook.

2 RELATED WORK

(S.Lane et al., 2011).conducted identified adaptation activities that could be used to adapt Service oriented system (SOS). These activities combined with a skeleton life-cycle model; proposed by the S-Cube consortium, formed the basis of reference process model' frame for adapting Service-Based applications (SBA). This frame of reference was used to guide interviews with development practitioners who had experience and could provide expert' opinion Service-Based applications' adaptation. Thecollected data of these interviews was transcribed and analyzed by using qualified content analysis techniques. The result of adaptation activities and tasks were constructed into a detailed

Elmagrouni, I., Kenzi, A., Lethrech, M. and Kriouile, A.
Adaptation Services-oriented Systems LifeCycle.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 223-231
ISBN: 978-989-758-187-8

process model identifying the relevant stakeholders and development artifacts for each stage of the process. The model's transfer and ability were demonstrated during an evaluation process where the model was systematically compared to a component-based application adaptation model and an empirically based SBA development life-cycle. The approach's advantages are over similar approaches because it focused and based on input provided by experts from the field.

A method proposed by Azevedo et al. [14] with activities to guide the designer to identify the most suitable set of services to support the business activities of the organization. The method consists of the following steps: (1) Selection of activities subject to automation - this stage selects process activities TO-BE where can be identified candidate services, (2) Process models are represented using the Event-driven Process Chains (EPC) and Function Allocation Diagram (FAD), (3) Candidate services identification and classification - activities identified in the previous step are analyzed within their contexts in process models according to a set of heuristics, and (4) Consolidation of candidate services supported by the use of heuristics.

A guideline is proposed by Shirazi et al. [15] for the service identification using two approaches: top-down and bottom-up. The bottom-up approach is used to identify applications and entities services; while top-down approach's goal is to recognize the business services and services oriented tasks. The method consists of the following steps: identifying business processes, making business use-case model, identifying entity-centric services, recognizing application services, identifying task-centric services and recognizing process centric services. Marks and Bell proposed a Service Lifecycle; this cycle includes the service evolution from conception to maturity along its execution. The identification of business services is performed using a top-down and bottom-up approach in iterative cycles. In order to identify new candidate business services, the author proposes an analysis of the following sources: business process, entities (interest and principal), budgeted projects, business experience, preexisting services and existing business applications. Arsanjani et al. [8] presented a method for service-oriented solutions developing called Service-Oriented Modeling and Architecture (SOMA). Specifically for services identification phase, the paper points out that a good practice uses a set of complementary techniques to identify services and cites three service identification techniques: (1) Goal-Service Modelling (GSM), treats the services aligned to the business goals, (2) Domain decomposition is performed through a top-down analysis of business domains and business processes modelling that are identified services, components and flows. The aim is to consider the static and dynamic view of the business including information rules and variations.

(3) Analysis of the existing asset is performed by bottom-upanalysis of the existing application portfolio and other assets and patterns that can be used to identify candidate services. After the application of the described techniques, the method also comprises the following step: refactoring and rationalization of the service whose service granularity is determined. Finally, a series of criteria are applied to determine which service is appropriate for candidates' publication.

3 E-TOURIST MOTIVATING SCENARIOS

Let us consider the following scenarios. Assume that a user would like to use a mobile device (e.g., PDA, iPhone, Smartphone, BlackBerry, etc.) with thedifferent operating system (IOS, Android, etc.) which is equipped with a GPS (Global Positioning System) to find relevant restaurants with/without open gardens. Figure 1 shows a model of application mobile "Restaurant Finder". Customers want to browse available restaurants to view offered food items and their cost. To this aim they must search a set of potential restaurants and select one among them. This application also aims to collect the customers' feedbacks regarding selected restaurants. Finally, the average satisfaction of all customers must be high. It can utilize various other services, such as the Reverse Geocoding for mapping GPS information to addresses, the Google Map for finding businesses close to an address, the Restaurant Data Service for searching restaurants based on the user preferences, and the Weather Information Service for obtaining weather information. The service also offers graphical information to the user with a reasonable delay (Wifi, 3G, and text/image).

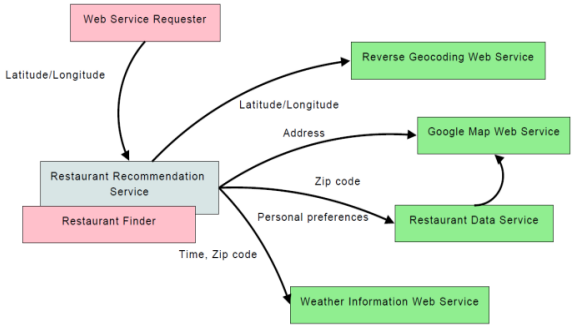


Figure 1: The Restaurant Finder.

4 ADAPTABLE SOS DEVELOPMENT PROCESS

The proposed method described in Figure 2 is based on MDA (Model driven architecture). It is composed of three abstraction layers: the CIM (Computation Independent model) which describes the system's requirements, the PIM (Platform independent Model) which specifies the system independently of any platform and the PSM (platform specific model) which contains the service models related to a specific platform.

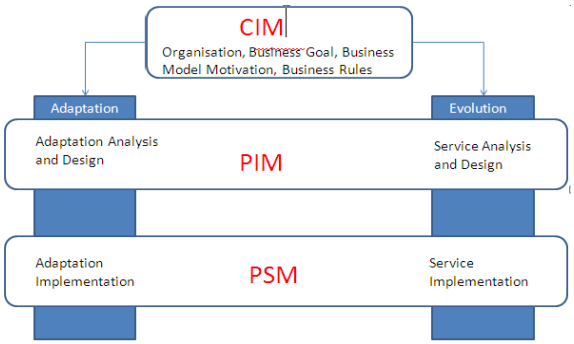


Figure 2: Development Process of adaptable Service Oriented System.

The first phase "Global analysis" is interested in the feasibility study.

Furthermore, the method is composed of two sub-processes:

An Evolution Process composed of the following activities: analysis and design service identify business requirement and implementation service.

An Adaptability Process which the main activities is: analysis and design Adaptation concerns the adaptation where system processes should adapt to certain conditions and implementation Adaptation.

4.1 Preparation

This phase is the preliminary study of the complicated organization that identifies two points: (1) business motivation model (2), Legacy system analysis.

4.1.1 Business Motivation Model

We define business requirements for business processes as the overall set ofrequirements that relate to business processes as given by the Business Motivation Model (BMM) of OMG, such as vision, mission, goal, strategy, objective andtactic. More specifically, a vision describes the future state of the application, withoutregard to how it is to be achieved, and mission indicates the ongoing activity thatmakes the vision a reality. A goal indicates what must be satisfied on a continuingbasis to effectively attain the vision, and a strategy is a long-term activity designed toachieve a goal. An objective is a specific and measurable statement of intent whoseachievement supports a goal, and a tactic is a short-term action designed to achieve anobjective.

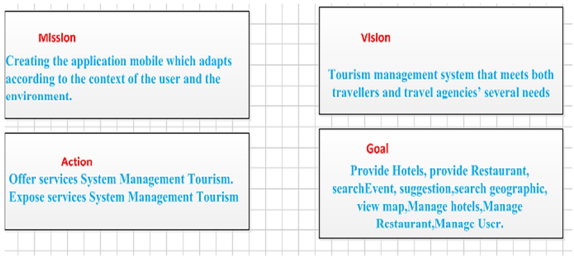


Figure 3: Business Model Motivation of Restaurant provider.

4.1.2 Legacy System Analysis

In this step, the decomposition of existing systems in the form of application modules is used to provide an implementation for business services that were previously identified. Then, we apply a bottom-up approach, i.e. starting from the existing system to the business services and business processes.

4.2 Services Analysis and Design

Business service identification is critical. Identifying appropriate services with the right level of granularity can have a major influence on the whole system. It is agreed that all business services should be identified to meet a business goal. Goals can be formulated at different levels of abstraction; ranging

from ahigh level and strategic to low level and operational.

4.2.1 Business Process Design with Goal Models

A goal model is used to capture why a business process is needed – its purpose or goal – and the different ways from which a goal can be attained. The goal model has been designed based on the design principle of Object-oriented Design, namely Decomposition and Abstraction. Decomposition principle is used to decompose a large problem into subproblems. Each sub problem is at the same level of detail, can be solved independently and can be combined to solve the original problem. Based on the above definition the goal model is designed to specify a high-level goal which is decomposed into subgoals and a hierarchical ordering of the subgoals is done. Goal tree structure is used to represent the model. In this, the high-level goal (problem) is decomposed into one or more subgoals (subproblems) and each subgoal is decomposed further into one or lower level subgoals. This goal becomes the root of the goal model. It is refined using AND/OR decompositions until the resultant subgoals can be delegated to either human actors or software services.

Figure 4 shows the goal of Restaurant Provider by tool OpenOME is an Eclipse-based tool designed to support goal-oriented, agent-oriented and aspect-orientedmodeling and analysis.

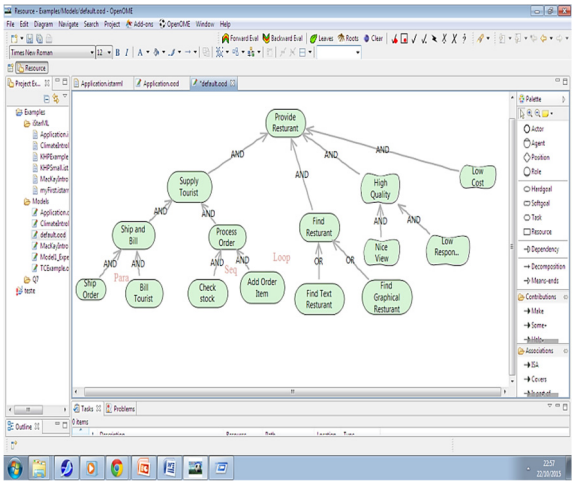


Figure 4: A goal model for the "Tourist System Management" business process.

The general goal is to provide Restaurants that AND-refined into the following sub-goals: Find Restaurant withhigh quality service and which provide both textual and graphical mode.

4.2.2 Enriching Goal Models for BP Modeling

Since we are interested in the automated execution of business processes, we needto capture more information about BPs than the basic goal models allow. A few annotationsare introduced for this purpose. Note that the annotations presented here arenot required to be formal. We use the following control flow annotations when employinggoal models to represent business processes:

• Parallel ("Para") and sequence ("Seq") annotations can be used with ANDdecomposedgoals to specify whether or not their subgoals are to be achievedin a temporal order. For example, billing customers and shipping goods are doneconcurrently in the process.

• Sel ("if(condition)") indicate the necessary conditions for achievingsubgoals. For example, in Fig. 2 the goal Order Out Of Stock Product isachieved only if the item is not already in stock.

•Loops ("while(condition)" or "for(setOfItems)"). For instance, the goal Add OrderItem must be achieved for all items in the order.

Table 1: Annotation for Business Processes.

Annotation	Meaning
Seq	Sequential execution of activities
Loop	Repeated execution of activities in a loop
Sel	Conditional selection of activities
Para	Parallel execution of activities with complete synchronization

4.2.3 Modeling of Input/Output

Modeling of input/output parameters of goals is also important for BP modeling. Identifying inputs and outputs during the analysis of a business domain help in determining resource requirements for achieving goals as well as for the sequencing of the goals. The types of inputs and outputs can also be specified. While optional, the input/output types can be used to generate detailed specifications for messages and service interfaces in a BP implementation.

Name: Collect Requests
Input/Output:r: ReceiveRequest; rc :RestaurantsCollection
DomainPrecondition:rc:state = default
DomainPostcondition:rc:state = req initialized
RequiredPrecondition:rc:keyword = "" ^ rc:date = null ^rc:distance = "" (r:keyword ≠ ""∨ r:date ≠ null∨ r:distance≠ "")

RequiredPostcondition:rc:keyword = r:keyword ^ rc:date = r:date ^ rc.distance=r:distance^Collect(rc:keyword;rc:date)
User: Tourist
Context: Device, Location, Time, Profile, Resturants Preference, Weather

Name: Find Graphical Content
Input/Output:rc :RestaurantsCollection
DomainPrecondition:rc:state = req_initialized
DomainPostcondition:rc:state = restaurants received
RequiredPrecondition:rc:keyword = "" ^ rc:date = null ^ rc:distance="" ^ (r:keyword≠ ""∨ r:date ≠ null∨r.distance≠ "")
RequiredPostcondition: ∀ *n* ∈ rc:restaurants: n:keyword = rc:keyword∨rc:date n:date∨ rc:distance= n:distance ^n:text≠ null ^ n:images ≠ null
User: Tourist
Context: Device, Location, Time, Profile, Resturants Preference, Weather

An operation is defined through name, input and output values and pre- and post-conditions. Required preconditions (ReqPre) define when the operationcan be executed.Required post-conditions (ReqPost) define additional conditions that mustbe true after execution. Domain pre- (DomPre) and post-conditions (DomPost) define the effects of the operation on the domain.The definition of operation Find Text Content is similar to operation Find Graphical Content except for the required post condition that is specified as follows:

ReqPost: ∀ *n* ∈ rc:restaurants: n:keyword = rc:keyword ∨ rc:date n:date∨ rc:distance= n:distance ^n:text ≠ null ^ n:images = null

4.3 Adaptation Analysis and Design

It is interested in capturingspecific adaptation requirement for SOS. To control service Adaptation, a designer needs to knowwhy a change was made, what are the implications, and whether the change is consistent or not. Eliminating spuriousresults and inconsistencies; that occur due to uncontrolled changes, is a necessary condition for services toevolve gracefully, ensure stability and handle variety on their behavior.

4.3.1 Typology of Adaptation

The nature of service Adaptations can be classifieddepending on their causal effects as follows:

- Minor Adaptations: these are small-scale incremental changes that are localized to a service or arerestricted to the clients of that service.
- Major Adaptation: these are large-scale transformational changes cascading beyond the clients of aservice, possibly to entire value chains.

Typical Minor Adaptation focuses on structural level changes (service types, messages, interfaces,andoperations) and business protocol changes (the conversations in which the service participates). TypicalMajor adaptation includespolicy-induced (pertaining to business agreements between service providers andconsumers.

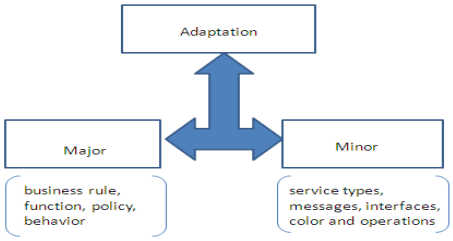


Figure 5: Minor and Major Adaptation.

4.3.2 Dealing with Adaptations

Service Adaptation requires an adaptation-oriented service lifecycle methodology to provide a sound foundation for spreading changes in an organized fashion that impacted services in a servicechain are appropriately configured, aligned and controlled as thechanges occur.

The purpose of the adaptation-oriented service life cycle is to ensure that standardized methods andprocedures are used for efficient and prompt handling of all servicechanges in order to minimize theimpact of change-related incidents upon service operation and quality. This means that in addition tofunctional (structural and behavioral) changes, adaptation-oriented service life cycle must deal with policy-induced operational behavior and non-functional changes. Figure 6 illustrates an adaptation-oriented service life cycle that comprises a set of inter-related phases, activities, and tasks that define the change process from the start through to completion. Each phaseproduces a deliveredmajor that contributes towards achieving change objectives. Logical breaks arealso provided in the change process and are associated with key decision points. The phases of the lifecycle are discussed in the next part.

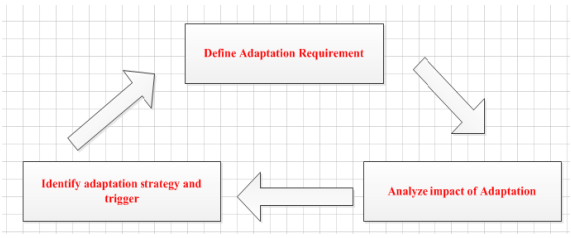


Figure 6: adaptation-oriented service life cycle.

The initial phase Figure 6 “Define Adaptation Requirement” focuses on identifying the need for adaptation and scoping its extent. One of the major elements of this phase is understanding the causes of the need for adaptation and their potential implications.

The secondin Figure 6 (“**Analyze impact of Adaptation**”) focuses on the actual analysis, re-designor improvement of existing services. The ultimate objective of service adaptation analysis is to provide an in-depth understanding of function, scope, reuse, and granularity of services that are identified for adaptation. The problem lies in determining the difference between existing and future service function.To analyze and assess the impact of changes, organizations rely on the existence of an “as-is” and a “tobe”service model rather than applying the changes directly to operational services. Analysts rely on an“as-is” service model tounderstandthe portfolio of available services. This model is used as thebasis forconducting a thorough re-engineering analysis of the current portfolio of available services that need toevolve. The “to-be” service model is used as thebasis for describing the target service functionandperformance levels after applying the requiredadaptation.To determine the differences between these two models, a gap analysis model is used to help prioritize, improve and measure the impact of service adaptation.

During the third and final phase **“Identify adaptation strategy and trigger”** in Figure 6, In order to select the adaptation strategy which should be applied, it is necessary to consider that adaptation maybe associated with a set of conditions and a trigger that are important for designing and performing adaptation. The trigger states when the adaptation Service must be activated. Each adaptation Service is operated through adaptation actions as explained in Table 2.

Table 2: Description of the two adaptation action.

Adaptation Action	Description
Substitution	The Possibility of configuration with a dynamic substitution of the service with another one
Performance	The possibility of going back in the process for performing an alternative path or redoing the same set of tasks

4.4 Service Implementation

It is based on the following step: Generating Business Processes

4.4.1 Generating Executive Business Processes

A method has been devised for using goal models to assist with the development and configuration of high- adaptability (flexible) BPEL processes. This makes it possible to generate BPEL processes that are easily readable by humans and are structured after the respective goal models. The BPEL code generation is semi-automatic. The generated code is not immediately executive so it needs to be completed. Nevertheless, provides valuable help in producing an executable BP based on the source goal model, the code is to be further developed by integration developers, who will also be selecting/designing Web services to be used by the process.

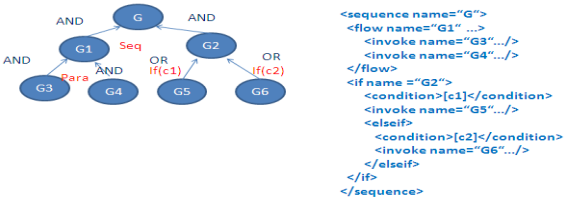


Figure 7: Example of WS-BPEL 2.0 code generation.

We start BPEL generation from the root goal and recursively through the goal tree until we reach leaf goals. Some of the goal models are presented to BPEL mappings through the example in Fig. 5, which shows a generic annotated goal model fragment. The root goal G has a sequential AND refinement, so it corresponds to the sequence operator in BPEL. G1 has a similar AND refinement, so it maps to the flow construct. G2 has a data-driven OR refinement, so it generates the if-elseif(BPEL 2.0) or the switch (BPEL 1.1) operator. Note that the conditions c1 and c2, which areinformal descriptions in the goal model, will be replaced with the appropriate conditions by a BPEL developer. While we abstract from some of the low-level BPEL details such as correlations, with the information captured in the annotated goal models, we also generate the following aspects of BPEL/WSDL specifications.

• We do an initial setup by defining the appropriate interface (portType), etc. for the process. A special portType for invoking the process and providing it with the (initial) configuration is also defined.

• A conditional/loop annotation for a goal G is mapped to the appropriate BPEL construct (e.g., if-elseif or switch, while, etc.) with the activity to be executed being the result of mapping the goal model subtree rooted at G into BPEL. The formal

conditions currently have to be specified manually.
• Leaf-level goals map into Web service invocations. The information in the goal model helps in defining the interface for the Web services invoked by the BP. We define appropriate WSDL messages based on input/output parameters of these goals. If data types are omitted from the goal model, they have to be supplied by a developer.
• Softgoals are used as the evaluation criteria in the configuration process and thus do not map into the resulting BPEL specification.

4.5 Adaptation Implementation

This activity involves in the implementation of the adaptation Mechanisms that were described in the phase analyze and design adaptation. Each conventional goal, which represents a functional requirement (i.e., it is operationalized),is mapped onto the corresponding sequence activity in the BPEL process. If the goal represents a non-functional requirement, but its nearest ancestor goal is operationalized, it is associated with the same sequence of its parent goal. This activity must represent an interaction of the process with its partner services (e.g., invoke, pick, and receive).Each adaptation goal is associated with a set of actions that must be performed at theprocess level.A triggering rule activates the evaluation of the trigger associated with the goal. A condition rule evaluates the conditions linked to the goal. If the two previous rules provide positive feedback, an activation rule is in charge of the actual execution of the adaptation actions. Performed when an adaptation goal can potentially fire (i.e., the corresponding Activation fact is available in the working memory) and is selected by the rule engine to be performed, among the other adaptation goals that can be performed as well. It executes the actions associated with that adaptation goal. For example, the triggering rule associated with ABG1 is the following:

```
when
Goal(id=="FGR", satisfaction < 1)
then
workmemory.insert(new
Trigger("TriggerABG1"));
```

5 ADAPTATION ASPECTS WEAVER

The adaptation Implementation follows a three-step process (see Figure 10):
1. Context detection consists of checking the runtime context information, in order to detect possible context changes. These tasks are performed by the Context Manager Service which is developed as a Web service in the BPEL process.
2. Aspect Activation is responsible for the plug-in and the removal of pre-defined Aspects into the BPEL process using the Aspect Activator Module.The Aspect Activator Module is conceived as an extension to the BPEL engine when running a process instance; the Aspect Activator receives the context change information from the Context Manager Service.
3. Updating original BPEL Process by activating the right Aspect which is executed in the BPEL process.

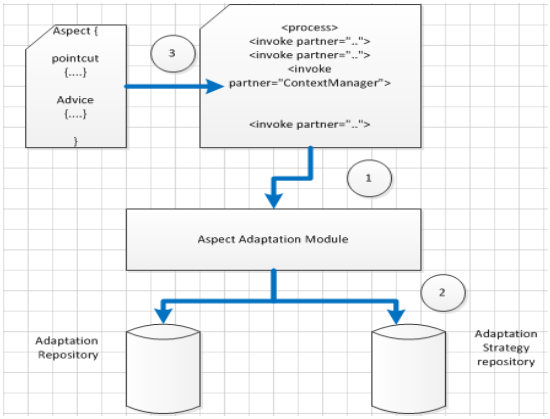


Figure 8: The adaptation process.

5.1 Adaptation Tools

We propose an evolved adaptationtool, based on the context manager and aspect, this tool allows selectingwhich services to invoke, and adapt them. These toolsare divided into three distinct layers, namely (see Figure 9): Application layer, Adaptation layer, and a Resource layer. These different layers are described as follows in the following.

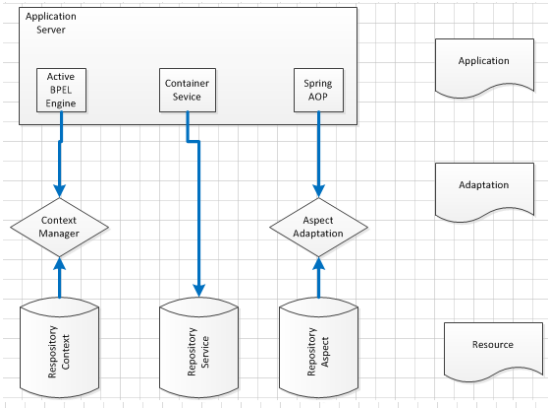


Figure 9: Technical architecture of adaptation.

Application Layer: It is the top layer of the technical architecture, including application platform which we implement our approach to adaptation. The central element of this layer application server. Typically, an application server is a server that Are installed applications used by customers.

Adaptation Layer: This is the second layer of this architecture. Placed between the application layer and resource, it contains components which will ensure the processing context information and any other operations required to carry out the adaptation of invoking services. Essentially, this layer contains two modules:

The context manager is charged to collect the information in context and to detect the possible changes of this information. The context manager will be called upon by process BPEL like any other web service.

The second module is the activation of aspect. Just like the context manager, the activation of aspect is implemented under the form of a Web service to interact better with the other elements of the architecture. This module is always placed after the context Manager in the process BPEL.

```
<?xml version="1.0" encoding="UTF-8"?>
<AspectRepository>

<aspect>
<aspect-identifier> Asp12</aspect-identifier>

<aspect-name> ExtraFees </aspect-name>
<condition> Delivering Place Casablanca</condition>

<joinpoint>ComputePrice(....) </joinpoint>jointpoint>
<advicetype> around </advicetype>

 <id-ws> 30 </id-ws>
</aspect>

</AspectRepository>
```

Figure 10: Aspect ExtraFes.

Resource Layer: It represents the third layer of this technical architecture. The resource layer includes all the resources needed for the other two layers. It interacts essentially with the matching layer at the two matching modules: the context manager and the aspect activation module. The resource layer includes the following:

- Repository Process: it lists all processes deployed to the BPEL engine.

-Repository Service: this case represents the functional service implemented as Web services.

-Repository Aspect: contains all aspects represented as an XML tree. Every aspect is a node in the tree characterized by an identifier and condition that represents a value of a context parameter.

-Repository Context: is supplied and updated by the context manager, the repository stores the context parameters related to the client and to the opportunity of cooperation.

6 CONCLUSIONS

In the previous sections, an approach is proposed to develop service-oriented systems that can be adapted to different contexts that deal with the effects of both major and minor. An adaptable-oriented service life cycle methodology is used to address and describe its phases. In particular, it discussed when a change in a service is triggered, how to analyze its impact and what are the possible implications of the implementation of the change for the service provider and consumers. A formal model for minor and majoradaptation,on the basis of the one, is the main goal of our future work.

REFERENCES

M.lethrech, I.Elmagrouni, A.Kenzi, M.Nassar and A.Kriouile“DSL and SOA, Exploratory Study” *JDTIC, 2012.*

H. Hafiddi, H. Baidouri, M. Nassar and A. Kriouile“A Model Driven Approach for Context-Aware Services Development”, in the *2nd International Conference on Multimedia Computing and Systems (ICMCSmodel'11),* Ouarzazate, Morocco, April 2011.

S.Lane,I.Richardson, A Process reference model developing adaptable service-based applications: *information and software Technology (2011).*

M. Shahrbanoo, M. Ali, and M. Mehran, “An Approach for Agile SOA Development Using Agile Principals,” *arXiv preprint arXiv:1204.0368, 2012.*

S.consortium,S-Cube knowledge model, 2011, http://www.s-cube-network.eu/km>.URL<http:/ /http:/ /www.s-cube-network.eu/km>

K. Boukadi, L. Vincent, P. Burlat, Modeling adaptable business service for enterprise collaboration, *Springer, Thessaloniki, Greece, 2009*, pp. 51–60.

A. Arsanjani, S. Ghosh, A. Allam, T. Abdollah, S. Ganapathy, K.Holley, SOMA: a method for developing service-oriented solutions, *IBM Systems Journal, 47 (2008) 377–396.*

M.P. Papazoglou,W.V.D. Heuvel, Service-oriented design and development methodology, *International Journal of Web Engineering and Technology, 2 (2006) 412–442.*

A. Kenzi, B. El Asri, M. Nassar, A. Kriouile, A model driven framework for multiview service oriented system development, *IEEE Computer Society, 2009, pp. 404–411.*

S.K. Johnson, A.W. Brown, A model-driven development approach to creating service-oriented solutions, *Springer, 2006, pp. 624–636.*

C. Canal, J.M. Murillo Software adaptation 12 (1) (2006).

I. Christou, S. Ponis, E. Palaiologou, Experiences of Using the Agile Unified Process in the Banking Sector, Software, *IEEE PP (2009) 1–1*.

A. Kenzi, B. El Asri, M. Nassar, A. Kriouile, A mdel driven framework for multiview service oriented system development, in: *IEEE Computer Society, 2009.*

C.Pahl, Semantic model-driven architecting of service-based software systems, *Information and Software Technology, 49(2007) 838-850.*

S. Mittal, Devs unified process for integrated development and testing of soa, University of Arizona 2007.

C. SooHo, A systematic analysis and design approach to. develop adaptable services in service oriented computing, in: *Congress on Services (Services 2007), pp. 375–378.*

Her, J., La, H., Kim, S.: A Formal Approach to Devising a Practical Method for ModelingReusable Services. In: Proc. *of 2008 IEEE Int'l. Conf. on e-Business Engineering.*

(ICEBE 2008), pp. 221–228 (2008).

S. Consortium, State of the art report on software engineering design knowledge and survey of HCI and contextual knowledge, Tech. Rep. PO-JRA- 1.1.1, July 2008.

Erl, T.: Service-Oriented Architecture: Concepts. Prentice-Hall, Englewood Cliffs (2005).

Cappiello C, Pernici B (2009) Design of repairable processes. In: *Cardoso J, van der Aalst W (eds) Handbook of Research on Business Process, Information Science Publishing.*

Bucchiarone A, Cappiello C, Di Nitto E, Kazhamiakin R, Mazza V, Pistore M (2009) Design for Adaptation of Service-Based Applications: Main Issues and Requirements. In: *Proc. of Fifth International Workshop on Engineering Service-Oriented Applications: Supporting Software Service Development Lifecycles (WESOA).*

A New Tool for Textual Aggregation In Information Retrieval

Mustapha Bouakkaz[1], Sabine Loudcher[2] and Youcef Ouinten[1]
[1]*LIM Laboratory, University of Laghouat, Laghouat, Algeria*
[2]*ERIC Laboratory, University of Lyon2, Lyon, France*
{*m.bouakkaz , ouinteny*}*@lagh-univ.dz, sabine.loudcher@univ-lyon2.fr*

Keywords: Aggregation, OLAP , Textual data, Algorithm.

Abstract: We present in this paper a system for textual aggregation from scientific documents in the online analytical processing (OLAP) context. The system extracts keywords automatically from a set of documents according to the lists compiled in the Microsoft Academia Search web site. It gives the user the possibility to choose their methods of aggregation among the implemented ones. That is TOP-Keywords, TOPIC, TUBE, TAG, BienCube and GOTA. The performance of the chosen methods, in terms of recall, precision, F-measure and runtime, is investigated with two real corpora ITINNOVATION and OHSUMED with 600 and 13,000 scientific articles respectively, other corpora can be integrated to the system by users.

1 INTRODUCTION

The huge increasing amount of complex data such as text available in different web sites, e-mails, local networks in business company, electronic news and elsewhere is overwhelming. This uncontrolled increase of information in the different fields, makes difficult to exploit the useful ones from the rest of data. This situation starts switching the information from useful to troublesome. The capability of OLAP tools available especially the text OLAP is not growing in the same way and the same speed the amount of textual documents is increasing. This problem is dramatically exacerbated by the big quantity of textual documents indexed by Search engines every moment. This makes the task of text OLAP and knowledge extraction from textual documents very limited and reduces the competitive advantage we can gain. Recently, a large number of systems have been developed over the years to solve this kind of problems and perform tasks in Information Retrieval; many of these systems perform specific tasks such as word counter and text summarization, however they are not in the level to satisfy the growing need of users to extract the useful information from documents using Text OLAP approaches.

In this paper we describe a software platform for keywords extraction and aggregation in an OLAP context. The platform implements a new way for extracting keywords from a corpus of document based on the Microsoft academia research web site and six algorithms for keyword aggregation which process a corpus of textual data to discover aggregated keywords.

The rest of the paper is organized as follows: Section 2 introduces related works in keywords extraction and aggregation in OLAP context. Section 3 describes the main components of the software prototype along with their functionalities. Whereas section 4 is devoted to numerical experiments. Finally, Section 5 presents conclusions and discusses further developments.

2 EXISTING APPROACHES AND TOOLS

Many approaches are proposed for keyword extraction but only a few for aggregation keywords. On the other hand, the majority of the existing work is based on information retrieval, and only some of them are in the OLAP context, where textual documents are stored in a data warehouse. In this section we make an inventory of the existing approaches in OLAP context, which describes a corpus of documents through the most representative aggregated keywords. There is a classical classification that includes the supervised and unsupervised approaches for keywords extraction, meanwhile in our case we introduce a new classification for textual extraction and aggregation approaches proposed in the OLAP context. We classify the previous works found in the literature into four categories. The first one uses statistical meth-

Bouakkaz, M., Loudcher, S. and Ouinten, Y.
A New Tool for Textual Aggregation In Information Retrieval.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 232-237
ISBN: 978-989-758-187-8

ods; the second one is based on linguistic knowledge; the third one is based on graphs; while the last uses external knowledge.

The approaches based on statistical methods use the occurrence frequencies of terms and the correlation between terms to extract the keywords. Hady *et al.* (Hady et al., 2007) proposed an approach called TUBE (Text-cUBE). They adopted a relational database to textual data based on the cube design, each cell contains keywords, and they attached to each keyword an interestingness value. Zhang *et al.* (Zhang et al., 2009) proposed an approach called Topic Cube. The main idea of a Topic Cube is to use the hierarchical topic tree as the hierarchy for the text dimension. This structure allows users to drill-down and roll-up along this tree. users discover also the content of the text documents in order to view the different granularities and levels of topics in the cube. The first level in the tree contains the detail of topics, the second level contains more general types and the last level contains the aggregation of all topics. A textual measure is needed to aggregate the textual data. The authors proposed two types of textual measures, word distribution and topic coverage. The topic coverage computes the probability that a document contains the topic. These measures allow user to know which topic is dominant in the set of documents by aggregating the coverage over the corpus. Ravat *et al.* (Ravat et al., 2008) proposed an aggregation function called TOP-Keywords to aggregate keywords extracted from documents. They used the $tf.idf$ measure, then they selected the first k most frequent terms. Bringay *et al.* in (Bringay et al., 2011) proposed an aggregation function, based on a new adaptive measure of $tf.idf$. It takes into account the hierarchies associated to the dimensions. Wartena *et al.* (Wartena and Brussee, 2008) proposed another method we called TOPIC in which they used the k-bisecting clustering algorithm and based on the Jensen-Shannon divergence for the probability distributions as described in (Archetti and Campanelli, 2006). Their method starts with the selection of two elements for the two first clusters. are assigned to the cluster of the closest of the two selected elements. Once all the terms are assigned, the process will be repeated for each cluster with a diameter larger than a specified threshold value. Bouakkz et al. (Bouakkaz et al., 2015) proposed a textual aggregation based on keywords. When a user wants to obtain a more aggregate view of data, he does a roll-up operation which needs an adapted aggregation function. their approach entitled GOTA is composed of three main parts, including: (1) extraction of keywords with their frequencies; (2) construction of the distance matrix between words using the Google similarity distance; (3) applying the k-means algorithm to distribute keywords according to their distances, and finally (4) selection the k aggregated keywords.

The approaches based on linguistic knowledge consider a corpus as a set of the vocabulary mentioned in the documents; but the results in this case are sometimes ambiguous. However, to overcome this obstacle, techniques based on lexical knowledge and syntactic knowledge previews have been introduced. In (Poudat et al., 2006; Kohomban and Lee, 2007) the authors described a classification of textual documents based on scientific lexical variables of discourse. Among these lexical variables, they chose nouns because they are more likely to emphasize the scientific concepts, rather than adverbs, verbs or adjectives.

The approaches based on the use of external knowledge select certain keywords that represent a domain. These approaches often use models of knowledge such as ontology. Ravat *et al.* proposed an other aggregation function that takes as input a set of keywords extracted from documents of a corpus and that outputs another set of aggregated keywords (Ravat et al., 2007). They assumed that both the ontology and the corpus of documents belong to the same domain. Oukid *et al.* proposed an aggregation operator Orank (OLAP rank) that aggregated a set of documents by ranking them in a descending order using a vector space representation (Oukid et al., 2013).

The approaches based on graphs use keywords to construct a keyword graph. The nodes represent the keywords obtained after pre-processing, candidate selection and edge representation. After the graph representation step, different types of keyword ranking approaches have been applied. The first approach proposed in (Mihalcea and Tarau, 2004) is called TextRank, where graph nodes are the keywords and edges represent the co-occurrence relations between keywords. The idea is that, if a keyword gets link to a large number of other keywords, this keyword will be considered as important. Bouakkaz et al. (Bouakkaz et al., 2014) propose a new method which performs aggregation of keywords of documents based on the graph theory. This function produces the main aggregated keywords out of a set of terms representing a corpus. Their aggregation approach is called TAG (Textual Aggregation by Graph). It aims at extracting from a set of terms a set of the most representative keywords for the corpus of textual document using a graph. The function takes as input the set of all extracted terms from a corpus, and outputs an ordered set, containing the aggregated keywords. The process of aggregation goes through three steps: (1) Extrac-

tion of keywords with their frequencies, (2) Construction of the affinity matrix and the affinity graph, and (3) Cycle construction and aggregated keywords selection.

The software system developed in this domain consists of two main components; Text Pre-processor and Topics Extractor. Text pre-processor, offers learning and inference functionalities. The learning functionality pre-processes a document collection by exploiting a stop words list and a general purpose to obtain the word-document matrix according to the bag-of-words model. The user can choose the number of words to be used for document indexing. The inference functionality processes a document to obtain one of the following bag-of-words representations; binary, term frequencies and the inverse term document frequency. Topic extractor implements a customized version of the Latent Dirichlet Allocation (LDA) model (Blei and Andrew, 2003). The solution of the LDA learning is obtained by using the Expected Maximization and the Gibbs Sampling algorithms which have been implemented in the C++ programming language on a single processor machine. Each topic is summarized through the estimate of its prior probability, a sorted list of its most frequent words together with the estimate of their conditional probabilities. Semantria [1] is a text analytical tool that offers an API that performs sentiment analysis and analytic text. Users can be integrated in the service to quickly yield actionable data from their unstructured text data, from review sites, blogs, or other sources. Additionally, users can download trial version and use Semantria for Excel, which installs directly into Office Excel to set up an environment for analyses.

3 THE SOFTWARE SYSTEM DESCRIPTION

In order to create a suitable environment for the online analysis of textual data, we intend to propose a new software which performs aggregation of keywords. The system described in this paper consists of three main components; namely Text Pre-processor, Keywords Extractor and Keywords Aggregator. These components have been integrated into a software system developed with Java programming language.

3.1 Text Pre-processor

This software component implements functionalities devoted to document pre-processing and document corpus representation. It offers words counter, and represents the documents of the corpus as a list of words with their frequencies (Figure1). Furthermore, binary and term frequency representations are allowed. The system takes the pdf, Microsoft Word and txt formats as valid inputs as shown in figure 1.

3.2 Keywords Extractor

This component is for keywords extraction. The keyword extraction function is based on the Microsoft Academic Search web site (MAS). MAS is a service provided by Microsoft to the public and it is free of charge. MAS classifies scientific articles into fifteen categories according to their fields. In each category it extracts the scientific keywords from articles and reorders them according to their frequencies. Our keywords extractor component uses this list of keywords and takes form each field the 2000 most frequent keywords, which are saved in separate text files. After that, Keywords Extractor process starts to compare MAS keywords with whole words extracted by the Text Pre-processor component. When a MAS keyword exists in the list, the extractor component saves it in a text file with its frequency and the name of the document in which it occurs.

Once our process is finished, we will get the right useful keywords validated by MAS. The output of this component is a two fold Matrix of document and keywords (MDKW). which is used by the third component to aggregate keywords.

3.3 Keywords Aggregation

The keywords aggregation component uses a set of textual aggregation algorithms TOP-Keywords, TOPIC, TUBE, TAG, BienCube and GOTA to aggregate keywords obtained in the previous step. it also produces the recall, precision, F-measure and the run time for each algorithm.

3.4 Graphical User Interface

The graphical user interface (GUI) is a necessary element in our system (OLAP-TAS) we take into consideration the ergonomical aspect to add an interactivity between the user and the machine when using our platform. The aim of the graphical user interface is to give the user a simple access to OLAP-TAS algorithms by a number of windows that help him to navigate in the system and test the different implemented algorithms without any need of previous Java programming experience or knowledge.

It is also helpful to assist students and researchers

[1] https://semantria.com/

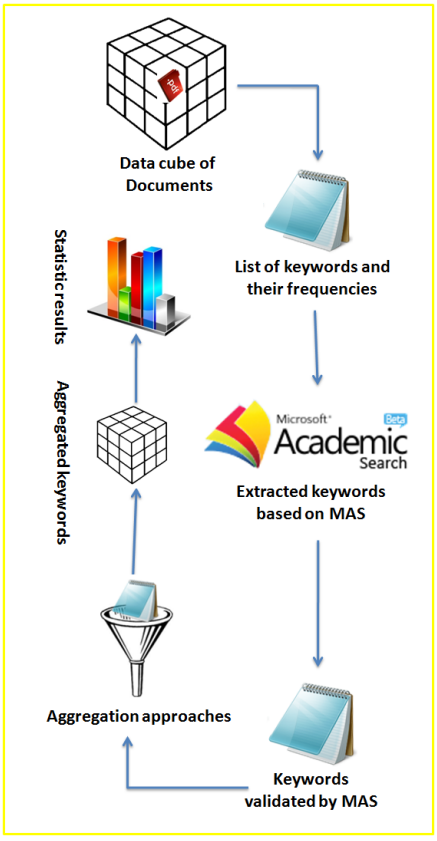


Figure 1: System architecture.

to do their scientific works and research experiments in a visual platform. It is obvious that the use of an interactive tool facilitates understanding and makes learning more beneficial task for many learners.

The GUI consists of two components: the first one is devoted to the preprocessing and keywords extraction and the second one is for Keywords aggregation. The Text Pre-processor and Keywords extraction components allow the user to create the *Documents* x *keywords* matrix based on Microsoft Academic Search web site (MAS) as shown in Figure 2. This interface gives users different possibilities to choose and configure the different parameters such as *Threshold* level and select the type of corpus (computer science, medicine, chemistry or all field of study). For the second interface which is devoted for Keywords Aggregation, it allows the user 1- to run, tests and compare the results obtained by the different implemented algorithms. 2- to visualize the aggregated keywords obtained by the different keywords aggregation approaches. 3- to compute different statistics for different approaches such as recall, precision, F-measure and run time, and save the different obtained results in various format *.xls*, *.txt* or *.doc* . 4- to change the corpus and run the Text Pre-

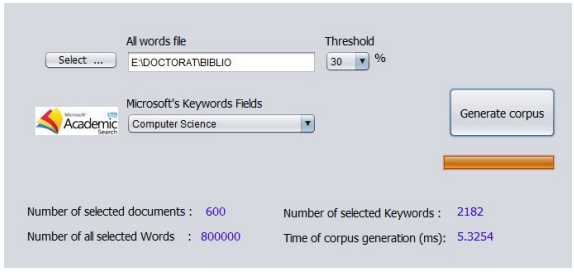


Figure 2: The Text Pre-processor and Keywords Extraction component interface.

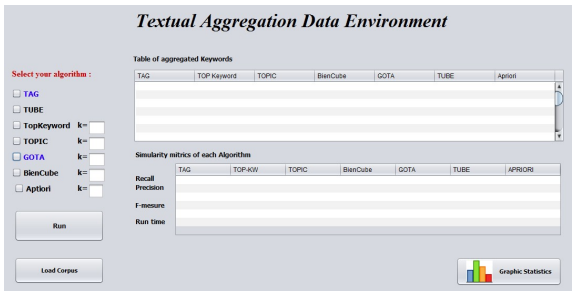


Figure 3: Keywords Aggregation component interface.

processor besides Keywords extraction components to load an other *Documents* x *keywords* Matrix, as shown in Figure 3.

4 RESULTS AND USAGES

4.1 Test and Results

In this subsection, we present an example to show how OLAP-TAS has been used. We compiled two real corpora, the first is from the *IIT* conference [2] (conference and workshop papers) from the years 2008 to 2012. It consists of 600 papers ranging from 7 to 8 pages in IEEE format, including tables and figures. The keywords are extracted from the full words according to the Microsoft Academia Search [3] keywords. The second corpus is used by many authors to test their works such as (Sebastiani, 2002) (Moschitti, 2003) (Moschitti and Basili, 2004), this corpus is called Ohsumed collection [4], it includes medical abstracts from the MeSH (Medical Subject Headings) [5], it contains 20,000 documents. In our case we selected 13,000 medical abstracts to test the performance of the implemented algorithm in our OLAP-TAS. For the evaluation task, many types of measures

[2] http://www.it-innovations.ae
[3] academic.research.microsoft.com/
[4] ftp://medir.ohsu.edu/pub/ohsumed
[5] http://www.ncbi.nlm.nih.gov/mesh/

have been proposed to evaluate keywords aggregation approaches, the majority of them insist on three measures, which are known as recall, precision, and F-measure. these measures are defined as fallows: The recall is the ratio of the number of documents to the total number of retrieved documents.

$$Recall = \frac{|\{RelevantDoc\} \cap \{RetrievedDoc\}|}{|\{RelevantDoc\}|} \quad (1)$$

The precision is the ratio of the number of relevant documents to the total number of retrieved documents.

$$Precision = \frac{|\{RelevantDoc\} \cap \{RetrievedDoc\}|}{|\{RetrievedDoc\}|} \quad (2)$$

The F-measure or balanced F-score, which combines precision and recall, is the harmonic mean of precision and recall.

To show the kind of results and statistics obtained by OLAP-TAS after the execution, we take the first corpus as an example to illustrate the different graphs obtained for different algorithms in Figures 4, 5, 6 and 7.

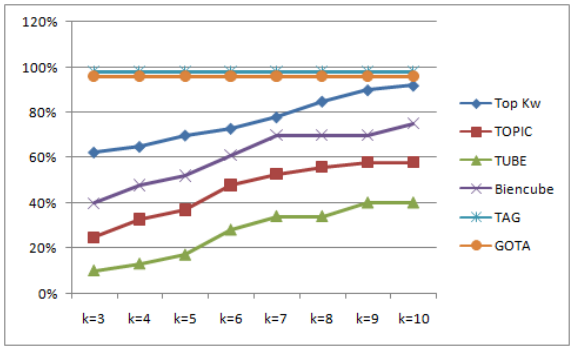


Figure 4: Comparaison of the Recall.

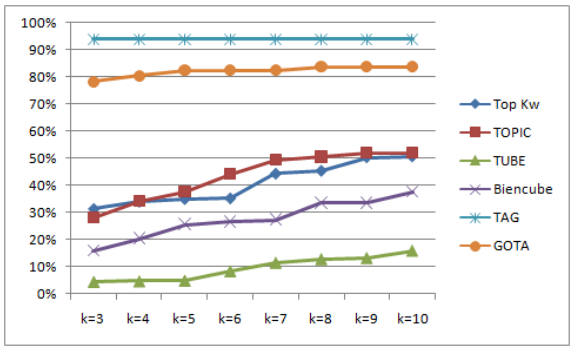


Figure 5: Comparaison of the Precision.

4.2 Uses of OLAP-TAS

In this section we will illustrate the use of the developed tool in both education and research.

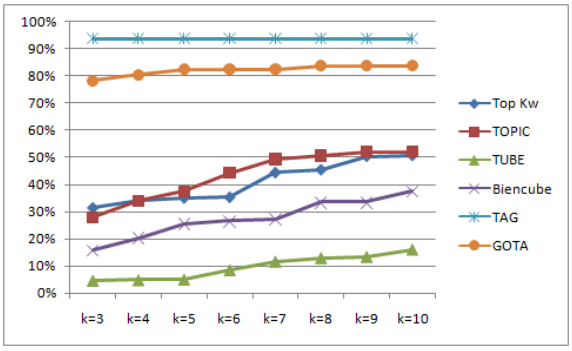


Figure 6: Comparaison of the F-measur.

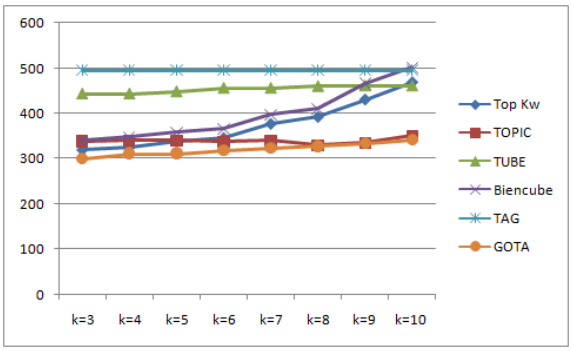


Figure 7: Comparaison of the Runtime.

Education: OLAP-TAS is a visual tool that instructors can use to help their students understand the basic concepts and the algorithms they face during their study. For example, it can be used to teach the students how the k-bisecting clustering algorithm based on the Jensen-Shannon divergence for the probability distribution works (Wartena and Brussee, 2008). As well as the $TF * IDF$ and their variation in Top-keyword (Ravat et al., 2008) and Biencube (Bringay et al., 2011). It can also help students to understand how to use graphs for textual by the selection of cycles in TAG (Bouakkaz et al., 2014) and the use of Google similarity distance (Cilibrasi and Vitanyi, 2007). In addition it shows the students how the recall, precision and F-measure change their values according to number of aggregated keywords k introduced by the user. Instructors may ask their students to do experiments with a real corpus using OLAP-TAS, write applications that use the Java classes, extend an existing approaches, or contribute in implementing a new algorithm to integrate in OLAP-TAS.

Research: OLAP-TAS contains implementations for several algorithms and approaches that solve common problems, such as textual aggregation in an OLAP context. It also comes with two corpora and annotated datasets. The implementation of other algorithms as well as other corpora, can be integrated into the plat-

form. This makes it a good resource for researchers to build systems and conduct experiments. OLAP-TAS was successfully used in several research projects as shown in (Bouakkaz et al., 2014).

5 CONCLUSIONS

In this paper a system for textual aggregation in text OLAP (OLAP-TAS) has been described. The software assists the user to discover the main aggregated keywords that best represent in a document collection. It is important to note that each approach is coded in a separate Java class to allow users to extend it or export it to another system. The use of OLAP-TAS reduces the amount of repeated code; it simplifies common tasks, and provides a graphical interface for textual aggregation approaches without requiring the knowledge in Java programming language.

REFERENCES

Archetti, F. and Campanelli, P. (2006). A hierarchical document clustering environment based on the induced bisecting k-means. *International Conference on Database and Expert Systems Applications*, pages 257–269.

Blei, D. and Andrew, Y. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, 42:993–1022.

Bouakkaz, M., Loudcher, S., and Ouinten, Y. (2014). Automatic textual aggregation approach of scientific articles in olap context. *10th International Conference on Innovations in Information Technology*.

Bouakkaz, M., Loudcher, S., and Ouiten, Y. (2015). Gota: Using the google similarity distance for olap textual aggregation. *17th International Conference on Enterprise Information Systems (ICEIS)*.

Bringay, S., Laurent, A., and Poncelet, P. (2011). Towards an on-line analysis of tweets processing. *Database and Expert Systems Applications*, pages 154–161.

Cilibrasi, R. and Vitanyi, P. (2007). The google similarity distance. *IEEE Transactions on Knowledge and Data Engineering*, pages 370–383.

Hady, W., Ecpeng, L., and HweeHua, P. (2007). Tube (text-cube) for discovering documentary evidence of associations among entities. *Symposium on Applied Computing*, pages 824–828.

Kohomban, U. and Lee, W. S. (2007). Optimizing classifier performance in word sense disambiguation by redefining sense classes. *International Joint Conference on Artificial Intelligence*, pages 1635–1640.

Mihalcea, R. and Tarau, P. (2004). Textrank: Bringing order into texts. *Empirical Methods in Natural Language Processing*, pages 26–31.

Moschitti, A. (2003). Natural language processing and text categorization: a study on the reciprocal beneficial interactions. *PhD thesis, University of Rome Tor Vergata, Rome, Italy*, pages 34–47.

Moschitti, A. and Basili, R. (2004). Complex linguistic features for text classification: a comprehensive study. *The 26th European Conference on Information Retrieval Research*, pages 34–47.

Oukid, L., Asfari, O., and Bentayeb, F. (2013). Cxt-cube: Contextual text cube model and aggregation operator for text olap. *International Workshop On Data Warehousing and OLAP*, pages 56–61.

Poudat, C., Cleuziou, G., and Clavier, V. (2006). Cleuziou g., and clavier v., categorisation de textes en domaines et genres. complementarite des indexations lexicale et morpho syntaxique. *Lexique et morphosyntaxe en RI*, 9:61–76.

Ravat, F., Teste, O., and Tournier, R. (2007). Olap aggregation function for textual data warehouse. *In International Conference on Enterprise Information Systems*, pages 151–156.

Ravat, F., Teste, O., and Tournier, R. (2008). Top keyword extraction method for olap document. *In International Conference on Data Warehousing and Knowledge Discovery*, pages 257–269.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, pages 34–47.

Wartena, C. and Brussee, R. (2008). Topic detection by clustering keywords. *International Conference on Database and Expert Systems Applications*, pages 54–58.

Zhang, D., Zhai, C., and Han, J. (2009). Topic cube: Topic modeling for olap on multidimensional text databases. *International Conference on Data Mining*, pages 1124–1135.

Semantic Integration between Context-awareness and Domain Data to Bring Personalized Queries to Legacy Relational Databases

Vinícius Maran[1,4], Alencar Machado[2], Iara Augustin[3] and José Palazzo M. de Oliveira[4]
[1]*Coordination Office, Federal University of Santa Maria, Av. Presidente Vargas, 1958, Cachoeira do Sul, Brazil*
[2]*Polytechnic School, Federal University of Santa Maria, Santa Maria, Brazil*
[3]*Center of Technology, Federal University of Santa Maria, Santa Maria, Brazil*
[4]*Institute of Informatics, Federal University of Rio Grande do Sul, Porto Alegre, Brazil*
{vmaran, palazzo}@inf.ufrgs.br, alencar.comp@gmail.com, august@inf.ufsm.br

Keywords: Database, Context-awareness, Ubiquitous Computing, Query, Semantic Web, Ontology.

Abstract: Context-awareness is a key feature in ubiquitous middleware. Mainly, it is applied to adapt services and interfaces of applications that use ubiquitous features. The application of context information to personalize data queries is a recent topic in computing and still presents a large number of challenges. One of the main gaps evidenced by this research field is the lack of integration between context information, which is designed and used by ubiquitous middleware, and domain data, which are frequently persisted in relational databases. This integration is necessary because context can be used as a filter for content query. This position paper presents a motivational scenario that clarifies the necessity of the integration between context, used by ubiquitous middleware, and relational data, a comparison between the state of the art of the field, a list of research opportunities in the field, and a proposal of a framework that uses ontologies to integrate context and domain data, modeled and stored in relational databases.

1 INTRODUCTION

Mark Weiser (1991) presented a series of scenarios where computing acts to help users in their daily tasks, without even they are able to notice the use of computers in these tasks. These scenarios originated the ubiquitous computing research field. Recently, ubiquitous computing area involves a set of technologies and methodologies of implementation.

One of the key concepts used in ubiquitous computing is called context-awareness. In recent definitions (Makris et al., 2013) (Perera et al., 2014), context information is defined as a measured and inferred data about current state of entities present in the environment. This information can be used in systems for adaptation in query of content and execution of services. Context is modeled and used in many forms by ubiquitous systems. Recent surveys (Bettini et al., 2010) (Strang; Popien, 2004) show that the representation of context based on ontologies presents a series of benefits over other forms of representation, like the existence of patterns to define ontologies and high expressiveness of them. Ontology can be defined as a formal and explicit specification of a shared conceptualization (Borst, 1997) and it can be represented in several languages. These languages are classified as: (i) based on logic, or (ii) serialized in XML languages. The second group is the most currently used because there are standards for representation, managed by W3C (2016).

Context information can also be applied in many forms in ubiquitous systems. The most frequent usage of context are (Dey et al., 2001): (i) dynamic adaptation of services, (ii) personalization of user interfaces, (iii) search of resources, and (iv) content and data querying. Content and data, related to the domain of the application, must be queried in a personalized form in ubiquitous systems, mainly using context information to filter it. This feature differs from the process of querying data in traditional systems in two ways. First, ubiquitous systems must query heterogeneous sources of data, including legacy relational databases. Second, the filtering information that is used in the query is not informed by the user explicitly, because the context information is automatically collected and inferred by ubiquitous middleware and then it is used in the query (Maran et al., 2015a). So for these domain data, persisted in relational databases, to be recovered in a contextualized manner, it is necessary to create forms of interconnection between context information, which are often represented in ontologies and domain

Maran, V., Machado, A., Augustin, I. and Oliveira, J.
Semantic Integration between Context-awareness and Domain Data to Bring Personalized Queries to Legacy Relational Databases.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 238-243
ISBN: 978-989-758-187-8

information, persisted in relational databases (Bolchini et al., 2013). This paper presents a motivational scenario of the data access based on context, an overview of the state of the art in the data access based on context research area, a list of research opportunities in the area, and a proposal of a framework to link legacy relational databases and context models, used in ubiquitous middleware.

The paper is organized as follows: In Section 2 a motivating scenario and the main concepts related areas of context-awareness, ontologies, and contextualized data querying are presented. In Section 3 a qualitative comparison between recent research and a list of research opportunities are presented. In Section 4 a framework to integrate legacy relational databases and context models based on ontologies is presented. In Section 5 the conclusions and future work are presented.

2 WHY UBIQUITOUS COMPUTING NEEDS TO GET CLOSER TO RDBMS?

Relational Database Management Systems (RDBMSs) traditionally support SQL queries. For a system make a query in the database, it is necessary that this system and the user inform in an explicit form the terms and conditions that will filter the results over the relations. This works well with traditional system and database design process, mainly because the database schema is designed to fit the application and domain information. But with recent advances in systems and the large increase of the information size in databases, some well-known problems became relevant about queries in RDBMSs field: **(i)** To perform a query is required to system and user to inform filtering criteria (the selection and projection criteria). A known problem is that often a few users report, or do not report filtering criteria. So the amount of information retrieved is large and users need to filter the information manually (Bolchini et al., 2013); **(ii)** The same instance of the database must be accessed by multiple systems, modelled in different manner, although of being designed for the same domain. This fact induces a known problem about semantics of the data, frequently supported by the use of ontologies (Dey et al., 2001).

Ubiquitous middleware frequently use ontologies to describe context models (Bettini et al., 2010). This way, applications that use ubiquitous features must use these context models to perform actions in a personalized form. These context information, previously measured and inferred by ubiquitous middleware, can be informed to the application or the database at the moment of the query (Perera et al., 2014). However, context information is not informed all in the same way every time. For example, in a specific moment, a body sensor can send some important information, but in other moment, this sensor cannot be available, and the context associated to this sensor cannot be used (Makris et al., 2013). This problem is related to the two previously mentioned problems in RDBMS queries (i and ii), mainly because the context information is not informed in a complete and in the same form in every query, and because the context is frequently modelled in ontologies. Furthermore, the design process of a relational database and context modeling occur in isolation from each other.

2.1 Motivational Scenario

Ubiquitous middleware have been designed and applied in several fields. Figure 1 presents an overview of the motivational scenario, described below. Recent researches (Stavropoulos et al., 2013) (Machado et al., 2014) describe smart university environments, which have a wide variety of educational resources that are managed by a ubiquitous middleware. Some of these resources can be recommended to students, which study in these universities (Stavropoulos et al., 2013).

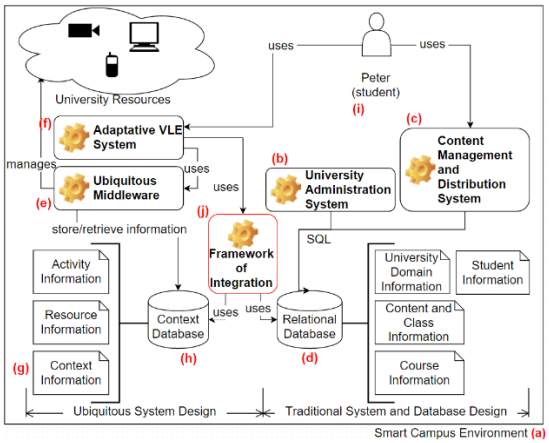


Figure 1: Motivational Scenario.

Let us imagine that a university campus **(a)** uses an university administration system **(b)**, and a content management system **(c)**, which allows teachers to provide supplementary materials, exercises and activities for students, which in turn access the system to get the materials. These systems store and retrieve information from a relational database **(d)**, using SQL

queries. In a determined moment, the university starts to use a ubiquitous middleware **(e)** with an adaptive Virtual Learning System (VLE) (Maran et al., 2015) **(f)** to show content and resources of the university to students. To adapt its execution and content to the user, the middleware manages the context information, captured from the environment **(g)**. To represent the context, the middleware uses an ontology, which represents the context after it being measured and inferred, and stores and retrieves this context from a database **(h)**.

Peter is a student **(i)** in an Electrical Engineering course in this smart campus. Currently, Peter is on the 2nd semester of the course and is attending the AL101EE discipline, called *Algorithms and Data Structures 1*. In a particular class, the context management middleware informs the context information about the user (1), about the educational context (2), information about location of the user (3), information about the device used by the user, and temporal and activity context information (5) to the adaptive VLE:

Context_User = {Peter, Student, 2ndSem, EEngineering} (1)
Context_Educational = {Peter,AL101EE,Class4,CBranch} (2)
Context_Location = {Country, City, SmartUniv} (3)
Context_Device = {AndroidBasedPhone} (4)
Context_Activity = {in_class, Init_time, End_time} (5)

This context information is modelled in the ontology, and it is used by the ubiquitous middleware. Currently, Massive Open Online Courses (MOOC) are offered in VLE **(f)** to an audience of the university as an opportunity to expand their knowledge. In a class of *Algorithms and Data Structures 1*, the teacher introduced the concept of Conditional Branch. At the end of the lesson, the recommender system showed to Peter in his smartphone the recommendation of the MOOC about algorithms. Peter is interested in more information about this topic, as recorded in MOOC.

By the time, the student enrols in MOOC, the ubiquitous middleware transfers context information (with the permission of the student) to the VLE system, and this information is used by the MOOC. Thus, profile and devices information associated with the student can be used by MOOC to filter course information to the student. The MOOC about Algorithms that Peter signed up presents algorithms and questions to assess student understanding on the basic algorithm structures applied to the student's course. Currently, the completion rate MOOCs varies between 5 and 10% (Pretz, 2014). Recent work (Pretz, 2014) (Quinn et al., 2014) attributed as a cause for this low completion rate the fact that the information related to courses are presented in the same way to all students, regardless of context information and information related to the student's profile. Originally, the MOOC platform was designed to use a relational database, with a pre-defined structure (EDX, 2015). As the context model is represented in ontologies and is managed by the ubiquitous middleware, it can not be informed explicitly at query time. Thus, there is a need to use up a framework **(j)** that implements a model that allows that context information used by the ubiquitous middleware to be used in information filtering. This scenario presents two relevant features regarding to recent research related to context:

- There is an exchange of context information between the middleware that manages context information on university campus and an external system (represented by the VLE). This is a research problem addressed in recent work (Makris et alk., 2013) (Perera et al., 2014);
- There is a gap for binding between context information and field data regarding persistence and recovery of data using context information in the query process. This gap is directly related to two characteristics: (i) information relating to the scope are usually modeled relationally and persisted in relational databases. This is evidenced by the wide adoption of RDBMS (DBEngines, 2016) and (ii) information related to context used by ubiquitous systems are often modeled on ontologies (Bettini et al., 2010). In recent work related to ubiquitous middleware, application independent ontologies were used as the basis for context modeling.

2.2 Context-awareness and Ontologies

Context is a broad term and has a set of definitions, according to specific fields. Context can be defined as *"any information that might be used to characterize the situation of entities that are considered relevant to the interaction between a user and an application"* (Dey et al., 2001). Context can be modeled in various ways. Some of the most common forms to represent context are: key-value pairs, object orientation model, logic based model, ontologies, and mark-up languages. According to Bettini et al. (2010), ontological models have greater capacity for representation and inference. Spatial models are more efficient if compared to ontologies and object oriented modeling, but they do not have as much representation capacity compared to the ontological models. This way, ontologies are the most used form to represent context information in ubiquitous architectures. Context-Aware Data Query states that the data retrieval and filtering operations should be based on context information reported to the system

time of information query. Thus, stored information, whether in structured or unstructured form, can be adapted according to the given context. As the context involves large sets of information, defined in fields, it can also be considered a document in a collection. Thus, contexts should be available for query in the same way that a common document related to the domain. Therefore, the context can be used in two ways in information retrieval (Bolchini et al., 2013): (a) To derive a query that returns the documents that best fit the required context; (b) To treat context as a document, i.e., the context becomes the source of information being queried.

Ontology is defined as a formal and explicit specification of a group of concepts in a shared form (Borst, 1997). As context model is a type of knowledge about the environment and the entities that composes the environment of the user and system, it can be represented in ontologies. Research have modeled context ontologies. These ontologies vary in multiple aspects, for example the number of defined concepts, domain of application, validation methods and language of representation. Rodríguez et al. (2014) conducted a comparison between ontologies that represent context information and activities. It was found that PiVOn Ontology (Hervás et al., 2010) was the ontology that better attends the comparison criteria. This ontology has been used in a set of works related to ubiquitous computing (Rodriguez et al., 2014). Context information are essential for ubiquitous systems because the treatment of this information and its use allows that ubiquitous system be able to adapt itself to the needs of users and other systems. These adjustments must be made in real time and can result in both application behaviour change and in information retrieval.

3 STATE OF THE ART

Recent research propose models and extensions of existing tools and models to integrate context models and domain data querying. HyConSC (Anderson et al., 2006) is a framework that allows context-based consultations to be integrated to applications that use relational databases. To realize the extension of queries, the framework uses its own context model, represented in graphical model. The context information are persisted as notes in documents. The Context-Relational Algebra was proposed as an extension of relational algebra, which supports a logical model that allows to integrate contextual information to relational databases (Martinenghi et al., 2009). A photo recommendation based on context tool was proposed by Viana et al. (2011). This tool uses semantic context information to recommend photos based on similarity calculation.

The CARVE methodology (Context-Aware Automatic View Definition over Relational Databases) was defined as a proposal for integration between relational databases and context information in the form of a process of automatic generation of views based on context (Bolchini et al., 2013). The implementation of the methodology is performed in a number of phases, some of which must be manually set. The context information is modelled in Context Dimension Trees. HARE (Time-Aware Location-Aware and Health-Aware Recommender) is a content recommendation application that uses time, location and health data of the patient to perform recommendations (López-Nores et al., 2013). To perform the content recommendation, ontologies were previously used to describe the metadata about that content. An architecture was proposed by Hahm et al. (2014) to perform the custom recovery of engineering documents based on analysis of user profile. To conduct a qualitative analysis on the state of the art in data query based on context, it was made a list of important features for analysis:

Context Model: The way that context information are modeled. It can be modeled in relational form (BDR), object-oriented (OO), based on logic (BL), graphical models (G) or ontologies (Onto); **Domain Model:** The way the domain information are modeled and integrated with context information. It can be modeled in relational form (BDR), as documents with semantic annotations (DAS), or as domain ontologies (Onto); **Integration Mode:** The way the integration of context information and domain data is made. It can be based on ontologies alignment (Alin), relational-algebra expressions (RA), or integration by algorithms by property analysis (Alg); **Query Language:** Data recovery can be defined by relational algebra expressions (AR), defined in SQL, SPARQL or SQWRL; **Database Model:** Some researches do not specify (NE) using DBMS, others use models like Triple Stores (TS) or Relational databases (BDR). Some researches do not use databases (NU). Based on the list set up to carry out analysis of related work. The result of the features analysis is shown in Table 1. As can be seen, none of the research models context based on a generic context ontology. This contributes to the context share issue (Makris et al., 2013) (Perera et al., 2014) between ubiquitous systems. Generic ontologies are often used in ubiquitous systems generally to adapt the execution of services and to infer contexts. However, there is the existence of a

Table 1: Qualitative analysis of state-of-the-art.

Features / Work	(Anderson et al., 2006)	(Martinenghi et al., 2009)	(Viana et al., 2011)	(Bolchini et al., 2013)	(López-Nores et al., 2013)	(Hahm et al., 2014)
Context Modeling	BDR	BDR	Onto	G	Onto	Onto
Context Modeling based on a generic ontology	-	-	-	-	-	-
Domain Modeling	BDR	BDR	Onto / DAS	BDR	Onto	Onto / DAS
Integration Mode	Alg	AR	Alin	AR	Alin	Alin
Query Language	SQL	AR	SPARQL	AR	SPARQL	SPARQL
Specific for a Domain	No	No	Yes	No	Yes	Yes
Database Model	NE	BDR	NE	BDR	TS	NU

gap for binding contexts and ubiquitous systems domain data (Perera et al., 2014).

Another important feature that was observed in relation to the related work is that none of the tools modeled context information in ontologies - to be the most complete and extensible, and integrated it with relational databases - the most widely used format for domain data, and persisted only in a database format. Thus, it is proposed in this paper a framework of integration and recovery of domain information based on context, modeled in ontologies, and relational databases, which represent domain information.

4 A FRAMEWORK TO INTEGRATE CONTEXT AND RDBMS

This paper presents a framework for domain information query. This information is modeled and persisted in relational databases, and context information are modeled on ontologies. This way, the model uses an ontology-based model to perform information filtering based on context information. So even without changing the structure of the database that represents the application domain information, systems can use context information modeled with high expressiveness for querying this information. Figure 2 presents a high-level view of the proposed model to integrate context modeling with relational data querying. The framework is divided in five main levels, named:

(a) External Entities: Ubiquitous middleware perform management of context and environmental resources. Even in these environments, external systems can use resources from middleware. As ubiquitous middleware manages context information, they may be required to inform the current context of the environment when performing a query. External systems in turn hold queries to data using the model, which returns a data set that comply with the filters informed by external systems, and context informed by ubiquitous middleware;

(b) Interfaces: To communicate with the framework, ubiquitous middleware use REST interfaces, informing the context through representations in JSON-LD language. The process of serialization of context ontologies in JSON-LD was previously presented in (Maran et al., 2015). External systems in turn carry the information query using SQL language;

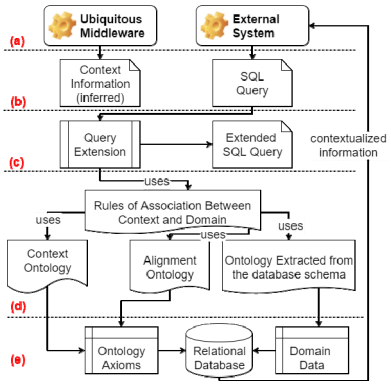


Figure 2: Overview of the framework.

(c) Query Extension: The query carried out by systems through SQL format is extended through a process. This process performs the query in association with rules that define links between context and domain data and checks for relationships that can be used to query. In this process, the definitions of ontologies are used to describe context, domain database schema, and the alignment of definitions of concepts;

(d) Conceptual Layer: This layer represent conceptual schemas and ontologies used by the model. In this paper, conceptual model is a modeling on a specific area, where this model was the basis for the logic model for a database that stores data about the domain. Some authors classify this type of conceptual modeling as a lightweight ontology (Maran et al., 2015a). To perform context-based information query, the model uses three distinct sets of settings: (i) an ontology that describes context based on PiVOn (Hervás et al., 2010), a context ontology independent of application, (ii) A conceptual

model that describes the domain database., and (iii) An ontology defining generic concepts and relationships to allow the alignment of context and domain information. The alignment of the ontologies was previously presented in (Maran et al., 2015);

(e) Persistence: Instances of conceptual models and ontologies used by the proposed model are persisted in a relational database. An initial implementation of the serialization of the definitions was presented in (Maran et al., 2015a).

5 CONCLUSIONS

Ontologies have been used by ubiquitous architectures for representing context information. Furthermore, inference rules have been used for making inferences about the context, which according to current definitions is measured and inferred knowledge about the status of entities.

Relational databases are used in most applications. As shown by motivating scenario, the context of use in the structured information retrieval in relational databases is relevant. In this work, an overview of the field was presented, as an study about the state of the art and the proposal of a model of integration between context, modeled on ontologies and domain information, modeled in relational databases. The framework is in implementation phase. As the future work, we pretend to evaluate it in a scenario based in the motivational scenario presented in this work.

REFERENCES

Anderson, K. M.; Hansen, F. A.; Bouvin, N. O., Templates and queries in contextual hypermedia. In: *Proceedings of the seventeenth conference on Hypertext and hypermedia. ACM, 2006*. p. 99-110.

Bettini, C., Brdiczka, O., Henricksen, K., Indulska, J., Nicklas, D., Ranganathan, A., Riboni, D. A survey of context modelling and reasoning techniques. *Pervasive and Mobile Computing, v.6*, n.2, p.161-180, 2010.

Bolchini, C; Quintarelli, E; Tanca, L. CARVE: Context-aware automatic view definition over relational databases.*Information Systems,v.38*,n.1,p.45-67, 2013.

Borst, W., Construction of engineering ontologies for knowledge sharing and reuse. Universiteit Twente, 1997.

DBEngines. Knowledge Base of Relational and NoSQL Database Management Systems. Website. Available at: http://db-engines.com/en/ranking.

Dey et al., A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human-computer interaction, v.16*, n.2, p.97-166, 2001.

Edx Website. Available at: code.edx.org/.

Hahm, G. J. et al. A personalized query expansion approach for engineering document retrieval. *Advanced Engineering Informatics, v.28*, n.4, p.344-359, 2014.

Hervás, R.; Bravo, J.; Fontecha, J., A Context Model based on Ontological Languages: a Proposal for Information Visualization. *JUCS, v. 16*, n. 12, p. 1539-1555, 2010.

López-Nores, M. et al. Context-Aware Recommender Systems Influenced by the Users' Health-Related Data. In: User Modeling and Adaptation for Daily Routines. *Springer London, 2013*. p. 153-173.

Makris, P.; Skoutas, D. N.; Skianis, C. A Survey on Context-Aware Mobile and Wireless Networking: On Networking and Computing Environments' Integration. Communications Surveys & Tutorials, *IEEE, v. 15*, n. 1, p. 362-386, 2013.

Maran, V., Palazzo M. de Oliveira, J., Pietrobon, R., Augustin, I. Ontology Network Definition for Motivational Interviewing Learning Driven by Semantic Context-Awareness. In Computer-Based Medical Systems, *2015 IEEE 28th I.S.* on. p. 264-269.

Maran, V., Machado, A., Augustin, I., Wives, L. K., & de Oliveira, J. P. M. (2015a). Proactive Domain Data Querying based on Context Information in Ambient Assisted Living Environments. In: *17th International Conference on Enterprise Information Systems*.

Martinenghi, D.; Torlone, R., Querying context-aware databases. In: Flexible Query Answering Systems. *Springer Berlin Heidelberg, 2009*. p. 76-87.

Machado, G. M.; Palazzo, J., Context-aware adaptive recommendation of resources for mobile users in a university campus. In: Wireless and Mobile Computing, Networking and Communications, 2014 *IEEE 10th International Conference* on. p. 427-433.

Perera, C., S. Member, A. Zaslavsky, e P. Christen. Context aware computing for the internet of things: A survey. Communications Surveys & Tutorials, *IEEE, v. 16*, n. 1, p. 414-454, 2014.

Pretz, K. "Low Completion for MOOCs". IEEE Roundup. Available at: http://theinstitute.ieee.org/ieee-roundup/opinions/ieeeroundup/low-completion-rates-for-moocs. 2014.

Quinn, S., Bond, R., & Nugent, C. D. "An Ontology Based Approach to the Provision of Personalized Patient Education". In: Ambient Assisted Living and Daily Activities. p. 67-74. *Springer. 2014*.

Rodriguez, N. D. et al., A survey on ontologies for human behavior recognition. *ACM Computing Surveys, v. 46*, n. 4, p. 43, 2014.

Stavropoulos, T. G. et al. aWESoME: A web service middleware for ambient intelligence. *Expert Systems with Applications, v. 40*, n. 11, p. 4380-4392, 2013.

Strang, T.; Linnhoff-Popien, C. A context modeling survey. In: *Workshop Proceedings. 2004*.

Viana, W. et al. Towards the semantic and context-aware management of mobile multimedia. *Multimedia Tools and Applications, v. 53*, n. 2, p. 391-429, 2011.

W3C Website. Available at: www.w3.org/.

Weiser, M. The computer for the 21st century. Scientific american, v. 265, n. 3, p. 94-104, 1991. Available at: http://doi.acm.org/10.1145/329124.329126.

Knowledge Management Framework using Enterprise Architecture and Business Intelligence

Oswaldo Moscoso-Zea[1], Sergio Luján-Mora[2], Cesar Esquetini Cáceres[3] and Norman Schweimanns[4]
[1]Faculty of Engineering, Equinoctial Technological University, Rumipamba y Burgeois, Quito, Ecuador
[2]Department of Software and Computing Systems, University of Alicante, San Vicente del Raspeig, Alicante, Spain
[3]Faculty of Systems Engineering, National Polytechnic School, Ladrón de Guevara E11-253, Quito, Ecuador
[4]innoCampus, Technische Universität Berlin, Straße des 17. Juni, Berlin, Germany
omoscoso@ute.edu.ec, sergio.lujan@ua.es, cesar.esquetini@epn.edu.ec, schweimanns@math.tu-berlin.de

Keywords: Knowledge Management, Enterprise Architecture, Business Intelligence.

Abstract: Knowledge Management (KM) has emerged as a tool which enables the efficient creation, use, distribution and transfer of knowledge in organizations. In the core of KM there are three dimensions of analysis: people, processes and technology. KM Frameworks presented in the past have had a strong theoretical background, but they have not been well explained in terms of how to implement them in practice to cover all KM dimensions. In this paper, a novel KM framework is presented. This framework was designed as a practical guide to implement KM endeavours in organizations. To accomplish our research objective, two management practices are incorporated in the framework: Enterprise Architecture and Business Intelligence. Enterprise Architecture allows companies to visualize organizational objects in different areas (business, applications and technology) through the use of models. Moreover, Business Intelligence technologies as data warehouses, data mining and visualization can enable the capture, transfer and the creation of new and purposeful knowledge. This work is intended to be a good resource for companies or individuals that want to implement a KM initiative.

1 INTRODUCTION

Knowledge Management (KM) has emerged as a discipline which enables the efficient creation, use, distribution and transfer of knowledge in organizations (Campbell, 2006). Innovations in science and technology have led to the emergence of intensively information-based organizations. These organizations need to transform this information into knowledge to secure competitiveness and improve decision making.

The core dimensions that need to be examined in a KM project are: people, processes and technology (Edwards, 2011). Knowledge derived from these dimensions should be analyzed and stored using different information repositories. A Knowledge Management Framework (KMF) enables organizations to conduct and implement KM initiatives. KMFs are the foundation for developing information infrastructure and information systems to manage knowledge properly. Karemente, Aduwo, Mugejjera, and Lubega (2009), describes different KMFs; however, none of these integrates and analyzes the three knowledge dimensions as a whole and are difficult to use in practice.

As a result of a university research project, a KMF was developed. This framework details how a KM implementation should be done in order to capture explicit and implicit knowledge derived from the three knowledge dimensions previously mentioned. Moreover, two management practices are included in the framework to accomplish our objective: Enterprise Architecture (EA) and Business Intelligence (BI).

EA is defined as "a coherent set of principles, methods and models that are used in the design, realization and maintenance of an enterprise's business architecture, organizational structure, information architecture and technology architecture with respect to the corporate strategy" (Lankhorst, 2009). The purpose of EA is to optimize the processes of an organization into a cohesive environment that is open to change and supportive to the business strategy (The Open Group, 2011).

On the other hand, BI is "the conversion of organizations resources to knowledge. It is the data mining and the integration of information from corporate data warehouses to produce large amounts

Moscoso-Zea, O., Luján-Mora, S., Cáceres, C. and Schweimanns, N.
Knowledge Management Framework using Enterprise Architecture and Business Intelligence.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 244-249
ISBN: 978-989-758-187-8

of information needed for effective decision making process and for planning strategically to achieve a competitive advantage in its industry" (Barakat, Al-Zu'bi, and Al-Zegaier, 2013). In this paper, a KMF supported by EA and BI is presented. The framework was designed as a practical guide to implement KM in organizations.

The rest of the paper is structured as follows: Section 2 presents the theoretical background; Section 3 explains how the KM framework was developed; and Section 4 provides conclusions of the work.

2 BACKGROUND

The research objective of this work is to present a KM framework which can be used in practice to capture, use and transfer knowledge. In this section, the literature research made for this work is presented.

2.1 Knowledge Management

Knowledge is one of the key resources that can strengthen the positioning of an organization (Curado, 2006). In order to sustain a competitive advantage, a resource should be valuable, rare and imperfectly imitable (Wernerfelt, 1984). Organizational knowledge meets these characteristics; therefore, it must be captured and managed appropriately. Knowledge can be defined as experience, facts, know-how, processes, beliefs, that increase an organizational or individual's capability (Karemente et al., 2009).

KM is "a process of identifying, capturing and leveraging the collective knowledge in an organization to help the organization compete" (Alavi and Leidner, 2001). Moreover, KM is "concerned with the exploitation and development of the knowledge assets of an organization with a view to furthering the organization's objectives" (Rowley, 2000). The reasons for KM include staff turnover, information overload, increasing need of expert staff, improved decision making and digitalization of organizational knowledge.

From the definitions, two important tasks are necessary to implement KM. Firstly, it is necessary to develop the technological infrastructure for facilitating knowledge capturing and sharing; and secondly, to establish mechanisms and procedures for retaining knowledge from people and processes. In order to accomplish these objectives, researchers have developed KMFs with different approaches. Nevertheless, a generally accepted framework has not been established (Heisig, 2009).

2.2 Enterprise Architecture

Enterprise Architecture (EA) supports in describing the current state (as-is situation) of an organization and proposes the best alternative solutions for the desired outcome (to-be situation). EA can be seen as a map that incorporates methods and techniques to create architectures in different layers of an organization. US Federal Enterprise Architecture Management Office defines EA as "a management practice to maximize the contribution of an agency's resources, IT investments, and system development activities to achieve its performance goals" (FEA Program Management Office, 2007).

EA addresses the need to manage increasing complexity and deal with continuous change by providing a holistic view of the organization, including their organizational components and their relations. EA is often viewed as a management practice that supports digitalization of knowledge to improve the performance of organizations (de Vries and van Rensburg, 2008).

Figure 1 shows a pyramid with the organizational architecture layers as: people, business, applications and technology. The circular arrows sequentially depict the process for implementing EA in an organization: getting the stakeholders involved, establishing management and control, defining the architecture process, the creation of the as-is and to-be scenario, development of a sequencing plan, using and maintaining the EA.

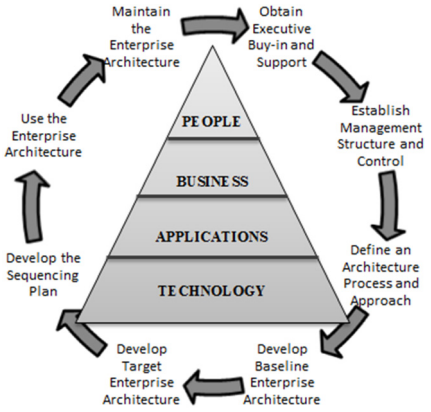


Figure 1: Enterprise Architecture Based on: (Tucker and Debrosse, 2003).

2.3 Business Intelligence

The term Business Intelligence (BI) was coined and became popular in the 1990s (Chen, Chiang, and Storey, 2012). According to (Gartner Inc., 2013), BI builds upon a set of tools and applications that

enable the analysis of vast amounts of information (Big Data) to improve decision making and performance of organizations. To accomplish this objective, decision makers require having access to all organization´s data, to analyze the business, its requirements and its trends.

The main technology in a BI project is a data warehouse. The data warehouse is a data repository which is populated from the integration of different operational data sources maintained in different units of the organization. An efficient analysis of data requires powerful analysis tools. Two main types of analysis tools exist: Online Analytical Processing (OLAP) and Data mining tools. OLAP tools use multidimensional views of aggregate data to provide access to corporate information for the purpose of improving decision making. Data mining uses software techniques for finding hidden patterns and trends in large databases to support strategic decisions (Connolly and Begg, 2005).

3 PROPOSAL OF KNOWLEDGE MANAGEMENT FRAMEWORK

As mentioned previously, in the core of KM there are three dimensions of analysis: technology, people and processes. Hence, a successful implementation of a KM initiative in organizations must take into account mechanisms to effective capture, use and transfer knowledge acquired from the three stated dimensions. The design of the framework is intended to put order in the KM process. Moreover, a practical framework can support managers in the creation, capture, digitalization of knowledge and decision making.

3.1 Technology

The first dimension of analysis in a KM process is technology. Technology is defined by (BusinessDictionary, 2015) as "The purposeful application of information in the design, production, and utilization of goods and services, and in the organization of human activities". In this paper, technology is referred to as objects used by humans (tools, software, hardware, machines) for KM. Information repositories for EA and databases are the core technologies that support KM.

EA repositories store the objects and processes modeled from the different architectural layers in an organization. On the other hand, databases store data generated from different applications. There are two main sources in which information can be found Online Transaction Processing (OLTP) and Online Analytical Processing (OLAP). The source of data for OLTP databases is operational data. The main purpose of OLTP databases is to run and to control fundamental business data with a highly normalized design. Data for OLAP databases is integrated and loaded from different operational sources into a multidimensional database namely a data warehouse. The purpose of OLAP databases is to improve business analysis and decision making. In a KM implementation, information can be extracted and processed from these two repositories. Many methods and techniques can be used to extract useful knowledge from databases. Some of the most used techniques in a knowledge discovery process are data mining and machine learning.

3.2 People

People dimension is one of the pillars for the exploration and exploitation of knowledge in organizations. According to Churchman (1975) "knowledge resides in the user and not in the collection of data". Thus, a mechanism should be designed within the proposed framework in order to capture and to transfer knowledge from people in organizations. It is important to note that the staff turnover rate in the United States in 2014 was 11% in all industries (Compensation Force, 2014). This is an indicator that strategies should be implemented to maintain and transfer knowledge from these and other groups of employees that are leaving organizations.

The cost of training of new employees without the efficient capture of people's knowledge can increase exponentially. According to the Association for Talent Development, the average of spending on employee training within US is around $1208 per year and per employee (Association for Talent Development, 2015). We believe that this value can be decreased if we plan staff turnover accordingly and establish mechanisms for the capture of knowledge with existing technology, for example by using learning management systems (Sanchez-Gordon, Calle-Jiménez, and Luján-Mora, 2015).

3.3 Processes

Processes are described by (Edwards, 2011) as "the way people, organizations and even technology actually do things". The importance of processes in KM initiatives are described in different papers (Bou and Sauquet, 2004) (Newell, Robertson, Scarbrough,

and Swan, 2002). Identification and digitalization of the core processes of an organization is an important step in a KM initiative. It facilitates the transfer of knowledge of tasks performed by staff since processes are divided into activities and procedures are created for easier interpretation. Processes are modeled normally in a Business Process Management (BPM) software or in an EA tool. The process models and architectures created in this software become an essential part of the knowledge base of the organization.

3.4 Proposed Knowledge Management Framework

A successful implementation of a KM initiative greatly depends on a well-defined method that supports the creation, capturing, use, distribution and transfer of knowledge. Organizational knowledge is created from different interdependent objects in different domains: strategy, product, services, information technology, applications, business processes and people (Lankhorst, 2009). Explicit and implicit knowledge can be derived from these domains. Explicit knowledge is knowledge that can be formulated, documented and reproduced. Implicit knowledge also known as tacit knowledge is knowledge that is difficult to document or formulate, and is normally associated with human knowledge.

Thus, the proposed framework intends to comprehensively create mechanisms to guide the KM process to capture knowledge from all the organizational dimensions. This framework was conceived as a part of a research project in a private university. The main goal of the research project is the design of a knowledge management framework (KMF) and the development of a web application prototype supported by databases, data mining and business intelligence tools for the planning process in the university.

One of the main objectives of the university is to position itself as a research and teaching institution, through the production, management and transfer of new knowledge based on institutional research lines. One of the projects implemented in the past year was the establishment of an institutional diagnosis in order to create a new model of corporate governance.

After analyzing the raised processes and the outputs of this project a need was identified. The identified need was to create a KMF for the planning area of the university to ensure the efficient management of knowledge and knowledge related activities. The purpose of the framework is to support planning, implementation and control of knowledge related projects and programs required for the effective management of intellectual capital.

Before the design of the framework started, a series of interviews was realized with different stakeholders in order to discover their knowledge requirements and to structure the framework. The importance of the three dimensions of knowledge was confirmed in the interviews. Moreover, certain activities to include in the framework were identified. Some of these activities were: discovering of knowledge in existing databases, digitalization of existing processes and the definition of mechanisms to convert tacit knowledge from different people in the organization into explicit knowledge. The novelty of the framework resides in the use of EA and BI to cover all the stated dimensions. Figure 2 depicts the designed framework.

The component in the right presents an analysis on how explicit knowledge is produced by using BI and EA tools. This box receives implicit knowledge as an input. The implicit knowledge is produced by people and processes in the organization. The knowledge discovery process inside the box has the following steps: analysis of existing databases and files, extraction of useful information, transformation to the target database format and loading. This process known as ETL (Extraction, Transformation, Loading) prepares data into a customizable format, cleans data with errors and eliminates duplicates. The purpose of this step is to load quality data into the target database in order to improve the analysis processes.

A data warehouse is the best target database for analysis. A data warehouse conceptual design consists of a set of dimension tables, fact tables and their relations. The populated data warehouse can be analyzed using BI and data mining tools to discover knowledge. Data mining and machine learning are popular methodologies for the knowledge discovery process. There are different methods and techniques that can be used.

On the other hand, digitalization of knowledge can be captured in an EA tool. An EA tool supports the creation of architectures to translate implicit knowledge into models which describe organizational structures (people), business processes, applications and technological infrastructure.

Most EA tools are based on the Archimate standard (Schekkerman, 2011). Archimate language allows the design of architectures in different domains and the creation of relations between the different objects of the organization. The

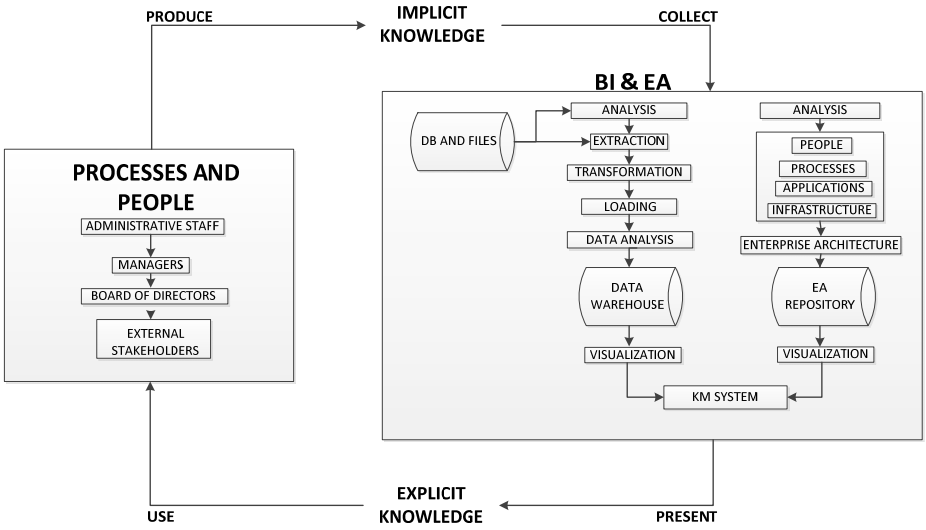


Figure 2: Proposed Knowledge Management Framework.

digitalization process includes an analysis of the different units and departments of the organization. Interviews must be realized with all the staff in order to document and model the different activities and processes realized in all architecture layers. The modeling of EA projects enables the capture and collection of implicit knowledge from the employees and the transformation to explicit knowledge in the forms of views and viewpoints of the architectures.

EA and BI are the main methodologies of creation of explicit knowledge. It is important to present explicit knowledge in an easy and understandable way. The framework suggests the presentation of knowledge by using a KM system which can be developed in a web environment. The results of the BI and EA process can be visualized and analyzed in this KM system. The output of the component in the right is explicit knowledge in the form of reports and dashboards that are presented to be used by people in all levels of the organization and can support in the design or redesign of new and existing processes. The explicit knowledge is the main input of the box in the left of the framework. Explicit knowledge can support and enhance decision making activities and can increase knowledge levels of the people in the organization. It supports as well the transfer of knowledge to new employees. As seen in the framework the KM process is a cycle in which knowledge is produced in a daily basis.

4 CONCLUSIONS

KM is a practice that organizations are incorporating to improve the creation, use, distribution and transfer of knowledge. The implementation of KM must be guided by a KMF. Many KMFs exist in the literature. However, these frameworks do not present practical mechanisms to gather and analyze all the knowledge dimensions: people, processes and technology.

The use of BI and EA tools bridges the gap of capturing all the knowledge dimensions. On the one hand, BI allows the transformation of simple information in valuable knowledge by applying data mining methods and techniques. On the other hand, EA supports the digitalization of implicit knowledge from people and processes by creating architectures in different domains. These architectures facilitate the transfer and distribution of knowledge to different levels of people in the organization. Some benefits of using this framework are: reduced training costs of staff turnover, improved decision making processes and the creation of a knowledge repository.

REFERENCES

Alavi, M., & Leidner, D. E., 2001. Knowledge Management and Knowledge Management Systems. *MIS Quarterly*, *25*(1), 107–136.

Association for Talent Development, 2015. 2014 State of the Industry Report: Spending on Employee Training. Retrieved November 20, 2015, from https://goo.gl/MpccrZ.

Barakat, S., Al-Zu'bi, H. A., & Al-Zegaier, H., 2013. The role of business intelligence in knowledge sharing. *European Journal of Business & Management*, *5*(2), 237–243.

Bou, E., & Sauquet, A., 2004. Reflecting on quality practices through KM theory. *Knowledge Management Research & Practice*, 35–47.

BusinessDictionary, 2015. Technology Definition. Retrieved November 20, 2015, from http://goo.gl/a266MR.

Campbell, H. M., 2006. The role of organizational knowledge management strategies in the quest for business intelligence. *Engineering Management Conference, 2006 IEEE International*, 231–236.

Chen, H., Chiang, R. H. L., & Storey, V. C., 2012. Business Intelligence and Analytics: From Big Data To Big Impact. *Mis Quarterly*, *36*(4), 1165–1188.

Churchman, W., 1975. The design of Inquiring Systems: Basic Concepts of Systems and Organizations. *American Educational Research Journal*, *12-1*, 94–96.

Compensation Force, 2014. 2014 Turnover Rates by Industry. Retrieved November 30, 2015, from http://goo.gl/hGEuFg.

Connolly, T., & Begg, C., 2005. *Database Systems*. Essex, England: Pearson Education Limited.

Curado, C., 2006. The knowledge-based view of the firm. *Instituto Superior de Economia E Gestao*, (1959), 18.

de Vries, M., & van Rensburg, A., 2008. Enterprise Architecture - New business value perspectives. *Southafrican Journal of Industrial Engeneering*, *19*, 1–16.

Edwards, J., 2011. A Process View of Knowledge Management : It Ain ' t What you do , it ' s the way That you do it. *Journal of Knowledge Management*, *9*(4), 297–306.

FEA Program Management Office, 2007. FEA Practice Guidance, (November). Retrieved from https://goo.gl/QIqI1V.

Gartner Inc., 2013. Gartner Business intelligence. Retrieved November 9, 2015, from http://goo.gl/LmJRG3.

Heisig, P., 2009. Harmonization of Knowledge Magagement-comparing 160 KM frameworks around the globe. *Journal of Knowledge Management*, *13*(4), 4–31.

Karemente, K., Aduwo, J. R., Mugejjera, E., & Lubega, J., 2009. Knowledge Management Frameworks. *Strengthening the Role of ICT in Development*, 35–57.

Lankhorst, M., 2009. *Enterprise Architecture at Work Modelling Communication and Analysis* (2nd ed.). Berlin Heidelberg: Springer-Verlag.

Newell, S., Robertson, M., Scarbrough, H., & Swan, J., 2002. *Managing Knowledge Work and Innovation* (2nd ed.). Palgrave macmillan.

Rowley, J., 2000. From learning organisation to knowledge entrepreneur. *Journal of Knowledge Management*, *4*(1), 7–15.

Sanchez-Gordon, S., Calle-Jiménez, T., & Luján-Mora, S., (2015). Relevance of MOOCs for Training of Public Sector Employees. In *14th International Conference on IT Based Higher Education and Training* (pp. 1–5). Caparica.

Schekkerman, J., 2011. Enterprise Architecture Tool Selection Guide. *Institute for Enterprise Architecture Developments*.

The Open Group, 2011. TOGAF ® Version 9.1. Retrieved from http://goo.gl/djuv15.

Tucker, R., & Debrosse, D., 2003. Enterprise Architecture Roadmap for Modernization. *Enterprise Modernization Issue*, *7(2)*.

Wernerfelt, B., 1984. A Resource-based View of the Firm. *Strategic Management Journal*, *5*, 171–180.

Business Opportunity Detection in the Big Data

Lyes Limam, Jean Lecouffe and Stéphane Chau
Altran Research, Région Sud Est, Division IIS, Altran, 1 Place Verrazzano, Lyon, France
{lyes.limam, jean.lecouffe, stéphane.chau}@altran.com

Keywords: Big Data, Business Intelligence, Text Mining, Graph Databases.

Abstract: Modern enterprise information systems are characterized by large amounts of data issued from various internal and external business applications, often stored and archived in different supports (databases, documents, etc. The nature of this data (voluminous, unstructured, heterogeneous, inconsistent, etc.) makes them difficult to use for analysis. In fact, it is typically an issue of big data analytics.
The main objective of our research project is to design a solution to detect opportunities (projects, new markets, skills, tenders, etc.) in the continually growing data, while adapting to its constraints. The extracted information should help users to take proactive actions to improve their business (e.g., identify a consultant skill that can be aligned with a given tender).
In this project we are interested in text data. There are two main reasons. The first reason is that text data is the most difficult to analyse by humans, especially when it is voluminous. The second reason is that we are convinced that valuable information is usually textual. Therefore, we define six research axes:

- Intelligent Information Sensing
- Text Mining
- Knowledge Representation (semantics)
- Querying the knowledge
- Results Interpretation
- Self-learning.

1 INTRODUCTION

Big Data mining is a recent and actual research trend (Diebold, 2012). Several approaches were experimented in several domains like: mobile communications (Laurila, et al., 2012), biology (Howe, et al. 2008), economics (World Economic Forum, 2012), (Letouzé, 2012), marketing (Fan, et al., 2015), decision making (Probst, et al., 2013), etc.

In Big Data mining, it is usual to deal with 3V problems: Volume, Variety, and Velocity. Recently, two more Vs were proposed: Variability and Value (Fan and Bifet, 2012). This last V is very important in a business-oriented mining, which has for objective to valuate internal and/or external data through mining.

In our case, the values we want to highlight are:

- Fast answer to customer's requests
- Understand customer's problems and determine new proposals to help solving it
- Find new business opportunities through new projects, new markets, skills, tenders, etc.

Some papers propose to use graph-oriented databases (Lin, 2014), which allow much faster responses than classical relational databases, due to local dependencies of data. A particular model of graph based on RDF (Resource Description Framework) retained our attention. RDF based model consists of "triples" [subject] -> [predicate] -> [object], which define conceptually a labelled graph (Bönström, 2003). That allows representing data dependencies in a clear, simple and efficient way, and allows fast access to data in graph-oriented databases.

One of the main problems is: how to build this graph? I.e. what are the data sources and how is extracted pertinent information. Some documents are relatively structured, like competencies records, but the most are non-structured and thus need to be threated specifically. Research axes to answer this question are crawling and text mining.

Another problem leads in graphs gathering: the objective is to build a global graph by combination of graphs extracted from documents. This problem is not trivial, as simply gathering graphs on their common nodes and links could lead to mistakes, mainly false

Limam, L., Lecouffe, J. and Chau, S.
Business Opportunity Detection in the Big Data.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 250-254
ISBN: 978-989-758-187-8

positive answers. This is a new research axe: the knowledge representation using graphs.

The interest of such a representation is to efficiently querying it. The issue of query formulation and searching in a global graph is not trivial. The objective is to propose a friendly-user interface for query building, and to give an understandable result: obviously, answer a sub-graph based on triple is not allowed for non-expert users; this becomes a problem of Human Machine Interface. This topic is covered by the axe of querying the knowledge.

After a query session, results have to be interpreted. A graphical representation with nodes and edges of the resulting graph is not adapted for a non-expert user. To allow a more efficient interpretation of the result it is needed to transform the resulting graph to a more user friendly representation. The research axe of result interpretation addresses this topic.

Last but not least, the results may be judged by the requestor more or less suitable. This feedback is necessary to increase efficiency of queries and accuracy of results. The last research axe called self-learning has the subject to consider how self-learning can be implemented in the query engine.

We choose to support this research a test case taken from an Altran's[1] need: how to match a business opportunity with the suitable consultant and in reverse, how to match a given consultant with a business opportunity.

When a customer asks our company for some skills, it is important to be able to find consultants that can answer its wishes in the best time. Internal data are very useful to retrieve consultants in the field of required qualifications: well-structured consultant skills and knowledge records, database of CVs that are also relatively well structured documents. But several external data sources are also very useful in skills search, like viadeo, linkedin, etc.

Matching request with the global graph will give us the possibility to have a look on nodes close to the request in order to retrieve additional information that can help to describe a context like new markets, skills, tenders, etc.

2 SYSTEM ARCHITECTURE

The main objective of Big Data technology is to integrate all data, then to analyze and represent this data in the unified schema. To reach this objective, we propose the following architecture (cf. Figure 1): This architecture is composed of three Parts:

Data Sources
It contains all data provided by Altran's tools. Data are extracted either from the enterprise databases or from flat files which can be structured or not structured.

These data were sorted and collected exhaustively to obtain relevant information that can be used to manage the Big Data System. The idea is to study and find a solution that allows to:

- Identify opportunities (projects, markets, skills, etc.)
- Improve transparency of existing data flow in the internal tools Altran
- Centralize existing information

Big Data Engine
The challenge of big data is to manage a large volume of data with optimal processing time.

We propose a big data engine based on Hadoop. Its HDFS file system allows the processing of very large amounts of data over several discs multiple machines as if it was a single storage volume.

In addition, the use of tools such as Storm can perform calculations and processing on graphs.

Implementing these two packages in Hadoop will reduce the time to treatment and process large volumes of data.

Other tools can also improve performance of process. For example, Kafka tool is used to enable the processing of queued messages.

Representation / Analyse / Compute:
The use of graphs databases (for example: Giraph, Neo4j) to represent the big data provides a unifying representation.

This offers a visual representation, easy to understand by the business. In addition, the use of graphs gives much better performance than relational databases, whether for the graph traversal or to load/import large data volumes.

3 OUR RESEARCH AXES

The objective of this section is to present the global framework of the solution to opportunity detection and its components which are the next research axis (cf. figure 2).

[1] Altran Technologies, SA is a global innovation and engineering consulting firm

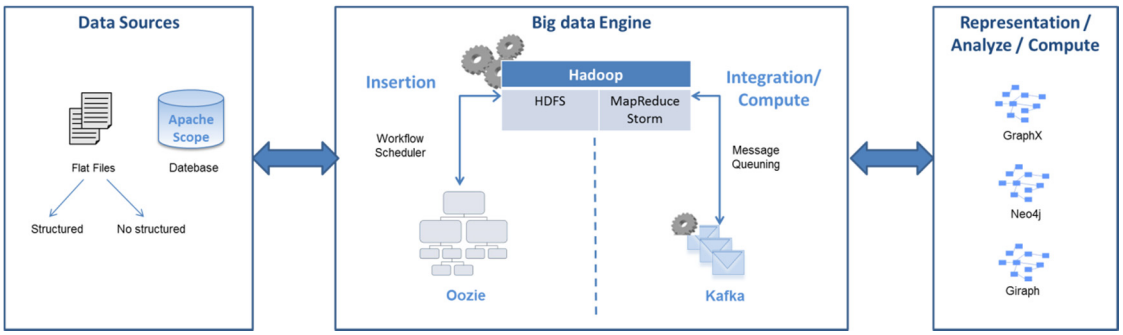


Figure 1: Representation of system architecture.

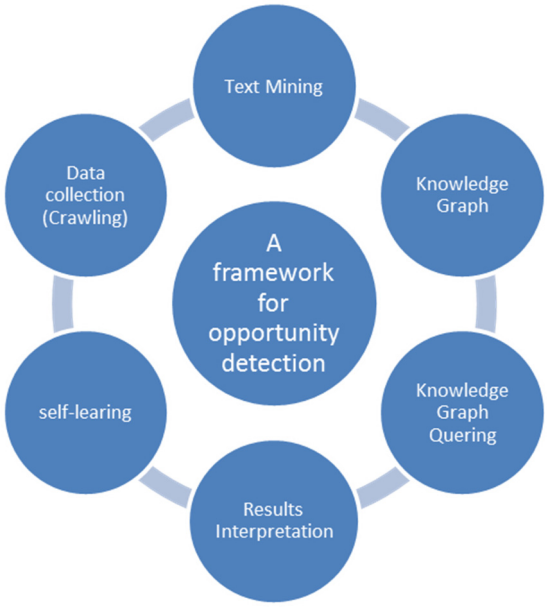


Figure 2: The Components of a Framework for Opportunity Detection.

3.1 Crawling

A crawler is software that explores recursively links found within a web page, from a pivot page, in order to collect and index the resources (web pages, images, videos, documents, etc.).

To enable the crawler to do its job correctly, one must define:

- A selection policy that identify the pages to download;
- A re-visit policy that defines when to check changes in the pages;
- A politeness policy that defines how to avoid overloads pages;
- A Parallelization policy that defines how to coordinate the crawlers in a distributed indexing.

With the introduction of the new Semantic Web research principles have been defined to allow crawlers to operate indexing methods involving more intelligent human-computer combinations as are practiced today.

The goal of this research axis is mainly to catch relevant pages in an intelligent manner. The ideal crawler should allow identifying pertinent data without drilling down the sources.

3.2 Text Mining

The crawling allows extracting from the WEB a set of sources which may have pertinent information. The second step is to inspect these sources to extract the information. It is the purpose of the text mining axis.

There are various types of sources that can be processed, but the large majority of them are textual: it is why we focus on text mining in this work.

Text mining is a complex process which deals with natural languages. The main difficulty is that a natural language is ambiguous, redundant and

implicit. Identifying new keywords and semantic links in a text need to use ontologies, heuristics and other sophisticated algorithms.

We choose to represent textual information as a global graph, where nodes are keywords, and edges are property links between these nodes. Edges are wearing semantic.

The nodes are found by keyword mining in the various structured and non-structured documents available in the company, like CVs, skill's records, etc. Using ontology based on business rules, enables us to categorize the identified keywords into different abstract levels, and to discover the semantic links existing between them. For example, the global graph should contain the consultants and their respective skills. This allows, for instance, retrieving the consultants that match a client request.

New keywords can be identified using a count of words and retaining only those which have a real sense. However for some structured documents like competency files, it is affordable to use the structure of the document to extract the relevant information. Anyway it is important that keyword extraction algorithms be able to adapt to the type of document.

Semantic links needs in many cases a certain degree of understanding of the analyzed document. This involves language treatments, using language tools enabling a more or less detailed analysis of the document content through content analysis techniques.

3.3 Graph based Knowledge Representation

As previously introduced, we choose to represent the extracted knowledge as an oriented and labeled graph where nodes are extracted keywords and links are semantic links between keywords.

Each analyzed document is resulting in a small graph representing the semantics extracted from the document. At this point we use RDF language to represent semantic graphs and to operate on them.

In order to form the global graph, we need to gather the different graphs coming from each analyzed document. As said before, this point is not trivial, because of the need to keep data dependencies from each to other; for example, assume A related to E related to B, and X related to E related to Y: a basic gathering will give A and X related to E related to B and Y, leading potentially to false positive answers A related to E related to Y, and X related to E related to B.

In order to deal with this constraint, we add an instance number in nodes: when such confusion may happen, the node is duplicated and a new instance number is given for each created node. In previous example, the node E is duplicated in two instances E[1] and E[2], thus we have A related to E[1] related to B and X related to E[2] related to Y; even if graphically the nodes are gathered, they are differentiated while analyzing the graph.

Another way may be to consider transitivity between links of different semantics, where A related to E and E related to B involves a potential transitive link A to B with a more precise meaning.

The global graph will first be built using well-structured documents like skills reports and CVs. It will be improved next by client requests and ontologies. Indeed, requests can enlarge application domains, add and refine skills, while ontologies will give abstraction levels that can extend or refine contexts.

3.4 Knowledge Graph Querying

Queries are given as small graphs similar to the global graph. A user interface will help to build queries in an understandable way, proposing refinements or contextual information that can give more precise formulations.

The graph of a query can be used, in active way, to quickly retrieve consultant that can satisfy a client request. It can be used also in a proactive way by augmenting the answer with neighborhood states or taking into account more general points of view, giving an extended view on client's needs, and thus allowing proposing complementary services. In a general active way, the sub-graph can be also used to determine market's trends, and identifying new relevant proposals of collaboration with new customers.

Some nodes of the query shall be "asking nodes": it will match any node of the global graph satisfying its relation links; thus, the entire query can match all the solutions of the need: for example, building a query on competencies with a "consultant asking node" will return all the consultants having these competencies.

An answer is thus a sub-graph deduced from global graph by matching query's nodes and links, with potentially several nodes for each "asking node".

Practically, the query graph construction could be not trivial: asking nodes could cover different information. For example, we could have consultants and enterprises skilled in programming languages: searching with an "empty asking node" will keep back consultants and enterprises. In order to reduce the field of answer, we decided to add a type to nodes; in such an example, consultants could be typed as person and enterprises as society. The query graph could then have a "typed empty asking node" in order

to retrieve only consultants.

Types are very important for query building, as we can propose to the user the list of types in which he can pick to refine its query.

Extracting the answer is not trivial too, as user could skip several intermediary nodes, because of not knowing it or simply to have a simpler query. Thus asking for a consultant skilled in java programming language could return a graph containing a node "object languages" that gather several languages like java, c++, c#, etc. More generally, simple queries could return complex chains of dependencies: the graph matching algorithm has to deal with this.

3.5 Result Interpretation

The query's result is a set of sub-graphs extracted from the global graph and matching the query graph. Representing this result in a human readable manner is not so trivial. It is easy to use a graphical representation where nodes are boxes and links are arrows. This form is acceptable for a human reading as long as the result set is not too big, but becomes unusable if there are hundreds or thousands of nodes.

Addressing this issue is not quite simple. Many studies have been done to try to solve it and many approaches exist (Shengqi et al., 2014), (Bergmann et al., 2014) with different approaches. However there is not any universal good approach. Each need may have own adapted representation. The goal is here to determine a good representation in existing tools, and if needed (and possible), adapt it to closely cover the specific need of our project.

3.6 Self-learning

Automatic learning concerns the design, analysis, development and implementation of methods allowing to a machine to evolve in a systematic process, and so fulfill the tasks difficult or impossible to fill by more conventional algorithmic means.

There are some kinds of self-learning algorithms. In our project we develop a self-learning method based on the user feedback. This method refers to a class of automatic learning problems, where the aim is to learn from experience, to optimize a quantitative reward over time.

The user feedback acts as a reward, and is used to improve the search algorithm, which in turn will be able to provide more accurate results.

4 CONCLUSION

Most of the techniques described above to achieve the goal of opportunity detection are recent research subjects. Some partial answers already exist, but it remains a lot of issues, difficulties and weakness in the big data mining and in the graph based knowledge representation. This research will try, using a test case of business opportunity detection, to address some of them and to propose original solutions to increase the efficiency and accuracy of the knowledge mining and restitution in the big data.

REFERENCES

Bergmann, G., Hegedüs, Á., Gerencsér, G., & Varró, D., 2014, 'Graph Query by Example', in CMSEBA in conjunction with MoDELS, pp. 17-24.

Ching-Yung Lin., 2014. 'Graph Computing and linked big data', Keynote speech at International Conference on Semantic Computing.

Diebold, F. X., 2012, '*A Personal Perspective on the Origin(s) and Development of Big Data: The Phenomenon, the Term, and the Discipline'*, Penn Institute for Economic Research, Department of Economics, University of Pennsylvania.

Fan, S., Lau, R. Y. & Zhao, J. L., 2015, 'Demystifying big data analytics for business intelligence through the lens of marketing mix', *Big Data Research*, vol. 2, no 1, pp. 28-32.

Fan, W. & Bifet, A., 2012, 'Mining big data: Current status, and forecast to the future', ACM SIGKDD Explorations Newsletter, Vol. 14, no 2, pp. 1-5.

Howe, A. D., Costanzo, M., Fey, P., Gojobori, T., Hannick, L., Hide, W., Rhee, S. Y., 2008, 'Big data: The future of biocuration'. *Nature*, Vol. 455, no 7209, pp. 47–50.

Laurila, J. k. et al., 2012, '*The Mobile Data Challenge: Big Data for Mobile Computing Research'*, Nokia Workshopp, in conjunction with Int. Conf. on Pervasive Computing, no EPFL-CONF-192489

Letouzé, E., 2012. *Big data for development: Challenges and opportunities*, UN Global Pulse.

Probst, L. et al., 2013, '*Big data Analytics and decision making',* Business Innovation Observatory, European Commission.

Shengqi Yang, Yinghui Wu, Huan Sun, and Xifeng Yan, 2014, 'Schemaless and structureless graph querying', *Proc. VLDB Endow.* Vol. 7, no 7 pp. 565-576.

Trelles, O., Prins, P., Snir, M. & C.Jansen, R., 2011, 'Big data, but are we ready?', *Nature*, Vol. 12, no 224.

Valerie Bönström, Annika Hinze, Heinz Schweppe, 2003. 'Storing RDF as a Graph', 1st Latin American Web Congress, pp.27-36

World Economic Forum, 2012. Big Data, Big Impact: New Possibilities for International Development.

INFORMATION SYSTEMS ANALYSIS AND SPECIFICATION

FULL PAPERS

Mixins and Extenders for Modular Metamodel Customisation

Srđan Živković and Dimitris Karagiannis
Faculty of Computer Science, University of Vienna, Vienna, Austria
{srdjan.zivkovic, dimitris.karagiannis}@univie.ac.at

Keywords: Metamodelling, Metamodel Composition, Metamodel Customisation, Metamodelling Tools.

Abstract: Metamodelling is a practical yet rigorous formalism for modelling language definition with a metamodel being its pivotal engineering artifact. A multitude of domain-specific modelling languages (DSML) are engineered to cover various modelling domains. Metamodels of such languages evolve over time by introducing changes and extensions and are further customised to suite project-specific needs. While majority of DSML development techniques provide concepts for creating metamodels from scratch, composition concepts for metamodel customisation beyond class inheritance are sought towards more flexibility and reuse. In this paper, we introduce a modular approach for metamodel customisation based on the idea of mixins and extenders. While mixins allow for defining self-contained metamodel modules for reuse, extenders enable non-intrusive composition of such reusable modules on top of existing metamodels. We show how this approach can be applied in a metamodelling tool such as ADOxx and demonstrate its usefulness by customising the BPMN language. The benefit of the modular metamodel customisation is twofold. On the language engineering level, our approach significantly promotes reuse, flexibility and overall efficiency in language definition and customisation. On the modelling level, the approach leverages engineering flexibility to provide custom modelling languages that better suits enterprise modelling needs.

1 INTRODUCTION

Model-based engineering approaches encourage the usage of modelling languages to analyse, design and develop increasingly complex systems and software. A multitude of standard and domain-specific languages are being engineered to cover various modelling domains. Independently of an application domain, modelling languages, like other software deliverables, evolve over time. New versions are released that introduce various changes and extensions (compare UML versions from 1.0 to 2.4.1 or BPMN versions from 1.0 to 2.0.2). Furthermore, released language versions are further adapted and customised to suite problem and project-specific needs. For example, a company may adopt BPMN 2.0 (OMG, 2013) as a standard for business process modelling, but it further requires company-specific extensions for process-based risk management. Such customisation may involve introduction of additional risk-related properties to existing language entities, creation of new entities or even integration with proprietary languages to build a custom hybrid solution. Ideally, such custom extensions should be portable to the upcoming version of the base language. The *evolving* nature of languages, the need for *customised* languages and the *complexity* that arises when combining evolution and customisation phenomena together, call for systematic, flexible and modular approaches for language design and customisation.

Metamodelling has been recognised as a practical yet rigorous formalism for modelling language development. In metamodel-based approaches, a metamodel is used to define the abstract syntax of the language. As a pivotal element in language definition, metamodel defines language concepts for which precise semantics and one or more concrete syntaxes may be defined (Selic, 2011). Nowadays, a multitude of mature *metamodelling languages* exist such as the standard MOF (OMG, 2014), or tool-specific meta-languages such as Eclipse EMF Ecore (Steinberg et al., 2008), MetaEdit+ GOPPRR (Kelly et al., 1996), ADOxx Meta^2-Model (Junginger et al., 2000; Kühn, 2010), or GME MetaGME (Ledeczi et al., 2001). In metamodelling languages, we may distinguish between *core* and *supporting* metamodelling capabilities. *Core constructs* are used to define fundamental elements of a metamodel. Constructs such as class, property or reference are examples of core constructs as they contribute to the core expressive power of a metamodelling language. Complementary, *supporting constructs* contribute to the efficiency

Živković, S. and Karagiannis, D.
Mixins and Extenders for Modular Metamodel Customisation.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 259-270
ISBN: 978-989-758-187-8

Table 1: Overview of core and supporting capabilities of selected metamodelling languages.

Capability	ADOxx Meta^2-Model	EMF Ecore	GME MetaGME	MetaEdit+ GOPPRR	MOF 2.0
Core capabilities					
Class	Class	EClass	Atom	Object Type	Class
Attribute	Attribute	EProperty	Attribute	Property	Property
Relation	Relation Class	EReference	Connection	Relationship	Association
Relation End	Endpoint	-	Connection Role	Role, Port	Property
Model Type	Model Type, Mode	EPackage	Model, Aspect, Role	Graph Type	Package
Supporting capabilities					
Modularisation	Library, Fragment	Package	Project	Graph Type	Package, Profile
Extensional Composition	Single Inheritance, Aggregation	Multiple Inheritance	Multiple Implementation Inheritance, Interface Inheritance	Single Inheritance, Inclusion	Multiple Inheritance, Package Merge, Stereotype, Extension, Tag

of metamodelling. They provide support for better structuring of metamodel artefacts and promote reuse of core metamodel artefacts. Modularisation constructs such as packages that are used to encapsulate metamodel elements, or composition constructs such as class inheritance which enables reuse of structural features of classes, are examples of such constructs.

While comprehensive support for the core metamodelling concepts is common to all metamodelling languages, the opposite is true for the supporting metamodelling constructs (see Table 1). Here, a positive exception, however, is the metamodelling standard MOF. Through the common UML2 infrastructure, MOF provides a set of mature mechanisms for metamodel customisation and incremental metamodel refinement such as the *Profile* mechanism and the *packageMerge*. Nevertheless, profiles have been, until now, exclusively used to refine only metamodels based on UML. In (Langer et al., 2012) the idea of UML profiles has been applied to Ecore, in order to enable profiles more broadly for DSMLs. Furthermore, while the package merge supports modular and incremental metamodel customisation, it operates on the level of packages and relies on the name-based element matching to apply merge, which is not always a desired approach for metamodel composition and customisation. Finally, the inheritance, in some of its forms as a single, multiple, interface or implementation-like, is supported by all metamodelling languages. Inheritance is most widely used technique for metamodel composition and customisation. However, while reusability of structural features by subclassing is one of the main advantages of inheritance, it may, at the same time, be its major drawback. Sublcassing as a way to extend a parent class with additional structural features may often end in complex class hierarchies and over-engineered metamodels. Furthermore, extending a class by subclassing may in some cases not be possible (single inheritance restriction, "sealed" base classes) or not desired (the base class is already in use, i.e. instances exist that would require tool recompilation and model migration).

Modular, incremental development has been one of the major drivers for the shaping of object-orient programming languages (OOPL). Single and multiple inheritance, mixins, traits, extension methods and templates in OOPLs are some of the key mechanisms that boost efficiency and flexibility in programming.

In this paper, inspired by some of the known extensibility concepts from OOPLs, we introduce a modular approach for metamodel customisation. In particular, the approach introduces the notions of *mixins* and *extenders* with appropriate composition operators that complement existing techniques for metamodel composition and customisation. Mixins allow for defining reusable self-contained metamodel extensions that can be combined by arbitrary modules without the creation of multiple class hierarchies. On the other hand, extensions allow for non-intrusive injection of custom metamodel extensions on top of existing metamodels eliminating the need for creating derived types. This way, with mixins we increase the overall potential for reuse in metamodelling beyond inheritance, whereas with extensions, we contribute to greater flexibility when extending metamodels. The paper is structured as follows. In Section 2, we introduce a running example related to the customisation of the BPMN metamodel. In Section 3, after we've revisited the limitations of inheritance, we introduce Mixins and Extenders and two new metamodel composition operators, Mixin Inclusion and Extension. Section 4 elaborates on the application of the approach based on the ADOxx metamodelling

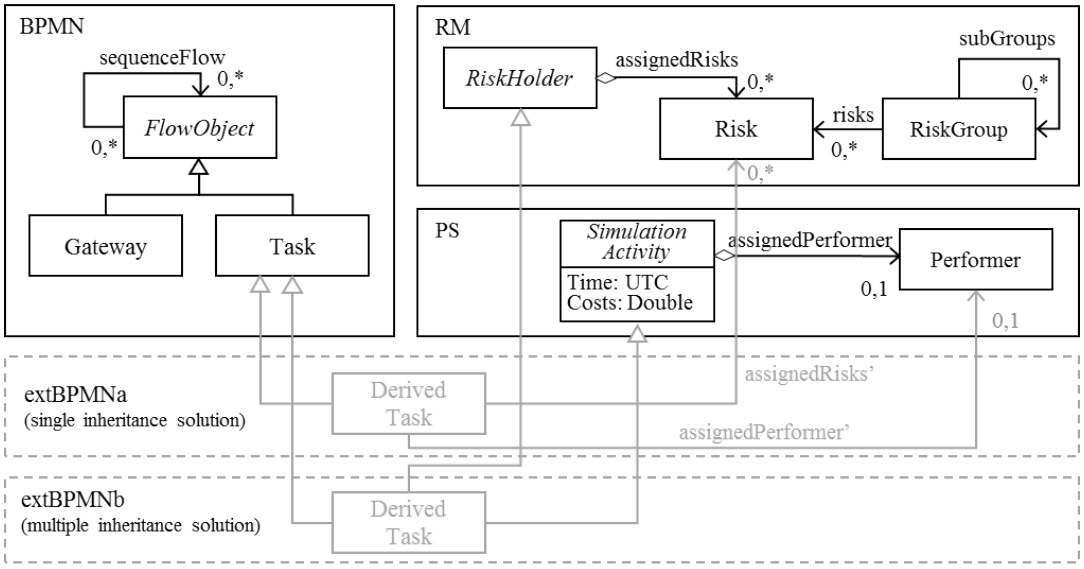


Figure 1: Customisation of the BPMN metamodel using reusable RM and DM modules.

language. In Section 5 we discuss the related work. Finally, Section 6 concludes the paper.

2 RUNNING EXAMPLE: CUSTOMISATION OF BPMN

Let us consider the previously mentioned example of metamodel customisation, in which, the industry standard business process modelling language BPMN 2.0 is extended by a business-oriented extension for process-based risk management (RM) and by a process simulation extension (PS) for business process simulation (Herbst et al., 1997). The necessity of extending the BPMN with further business aspects for enterprise modelling and analysis has been discussed in (Rausch et al., 2011).

Let us suppose we follow a modular approach to metamodel development, where metamodels are encapsulated into reusable, stand-alone modules, focusing on single aspects and concerns. In that case, we will have three metamodel modules, *BPMN*, *RM* and *PS* as depicted in Figure 1. In module *BPMN*, the class *Task* is the central entity for modelling business process flows, which we aim at extending with other related concepts. In module *RM*, the class *Risk* is used to model company risks. A risk may be assigned to various business entities, which is represented by the abstract class *RiskHolder*. In the *PS* module, the abstract class *SimulationActivity* represents an abstract activity containing attributes *Time* and *Costs* necessary to run the process simulation algorithm. Note that this is a very simplified view of the simulation aspect. Such activity may have an assigned performer (*Performer*) that executes the activity during the simulation. Our goal is to customise BPMN in a way that the task becomes connectable to risks, and that it contains simulation features. In other words, we want from class *Task* to have characteristics of both classes, *RiskHolder* and *SimulationActivity*. Furthermore, metamodel composition should be *non-intrusive*, i.e. both BPMN as well as RM and PS modules must not be modified. In addition, no new derived entities should be defined, in order to retain the compatibility with existing mechanisms and model bases.

3 METAMODEL COMPOSITION BASED ON MIXINS AND EXTENDERS

In this section, we first discuss current limitations of the inheritance in metamodelling languages based on the running example, since inheritance, in some of its variants, is commonly supported composition mechanism by all metamodelling languages (see Table 1). Afterwards, we introduce the concepts of Mixin and Extension, two new metamodel composition operators for flexible, modular metamodel customisation.

3.1 Inheritance is Not Enough

In a nutshell, the intention of inheritance is to reuse structural features of classes such as properties and references by creating parent-child class hierarchies.

A subclass inherits all features of either one super class (single inheritance) or of more than one super class (multiple inheritance). While metamodelling languages such as Ecore, MetaGME and MOF support multiple inheritance, languages such as ADOxx and GOPPRR are restricted to single inheritance. Multiple inheritance has been discussed controversially since its introduction in programming languages (Bracha, 1992), as well as, more recently, in metamodelling (Selic, 2011). Multiple inheritance is criticised for an increased unanticipated complexity and ambiguity in class design, allowing for antipatterns such as "diamond inheritance problem" and over-generalisation.

Figure 1 illustrates how both single and multiple inheritance can be applied to extend the *BPMN* module with *RM* and *DM* modules. We summarise major deficiencies in the context of customisation in two categories, *singleness* and *subclass imperative*.

Singleness. Given the single inheritance restriction by a metamodelling language, we introduce new extension module *extBPMNa* which contains one derived class *DerivedTask* as a subclass of *Task* from module *BPMN*. Since multiple inheritance is not allowed, we cannot inherit additionally from classes *RiskHolder* and *SimulationActivity*, but need to define two new references *assignedRisks'* and *assignedPerformer'* to classes *Risk* and *Performer* and to remodel properties from the class *SimulationActivity* such that the class *DerivedTask* includes the semantics of classes *Risk* and *Performer*. Obviously, this approach is not flexible enough as it doesn't allow for reuse of additional structural features other than those that are inherited from the single super class. If a metamodelling language allows for multiple inheritance, the class *DerivedTask* in module *extBPMNb* may specialise the class *Task* and also inherit from the abstract classes *RiskHolder* and *SimulationActivity* to accompany all required features. This solution appears to be more elegant, as it allows for reuse by inheritance from multiple superclasses.

Subclass imperative. Although multiple inheritance overcomes the problems of singleness, in both inheritance-based solutions, however, we are forced to introduce an explicit derived type in order to extend a class without directly modifying it. In our case, the class *DerivedTask* inherits in both cases from the class *Task* and must be used if extended process modelling with risks and simulation is desired. This kind of customising by subclassing may be an undesirable when applying metamodel customisation. It forces the introduction of a new modelling class in the language, even though, conceptually, only an adaptation of an existing class was required. Furthermore, as an effect of a new derived type, both functionality and models conforming to the base BPMN metamodel have to be upgraded to the new metamodel version, i.e. the instances of class *Task* need to be converted to the subclass *DerivedTask*, in order to apply the extension. We call this problem the *subclass imperative deficiency*.

3.2 Mixin-based Metamodel Composition

In order to mitigate the singleness problem of inheritance and complex and potentially ambiguous multiple inheritance hierarchies, while increasing the potential of reuse, we propose the usage of mixins in metamodelling. Adopting the general idea of mixin-based inheritance (Bracha and Cook, 1990) in metamodel composition, mixins are said to allow for the definition of independent metamodel element parts (Mixins) that may be reused, i.e. *mixed*, by other elements. Mixins usually bundle some common set of features that may be shared among other metamodel elements. To allow for the mixin-based metamodel composition, a *parent element*, a *mixin metamodel element* and a *mixin inclusion composition operator* are needed.

- *Parent element*. A parent element in the mixin composition may be any element of a compound type, i.e. an element that contains other elements. For instance, a class is a compound metaclass that may contain structural features such as properties and references.
- *Mixin element*. A mixin element is a compound element type that contains features to be shared among other elements. We define mixin element as a non-instantiable, abstract element, in order to denote its pure *supporting metamodelling capability*. The mixin element must be of the same type as the base element. For example, an abstract element of type Class may be defined that contains a set of common properties and/or references that may be shared between various other classes. In our case, appropriate candidates for mixin classes are *RiskHolder* and *SimulationActivity*.
- *Mixin inclusion*. Mixin inclusion operator is a relation that takes a parent element and a mixin element as an input, and includes ("mixes in") the child elements of the mixin element to the parent element. A parent element may mix in many mixin elements. In turn, a mixin element may be reused by arbitrary parent elements. Hence, mixin operator allows for an increased flexibility in appending features to an element without dis-

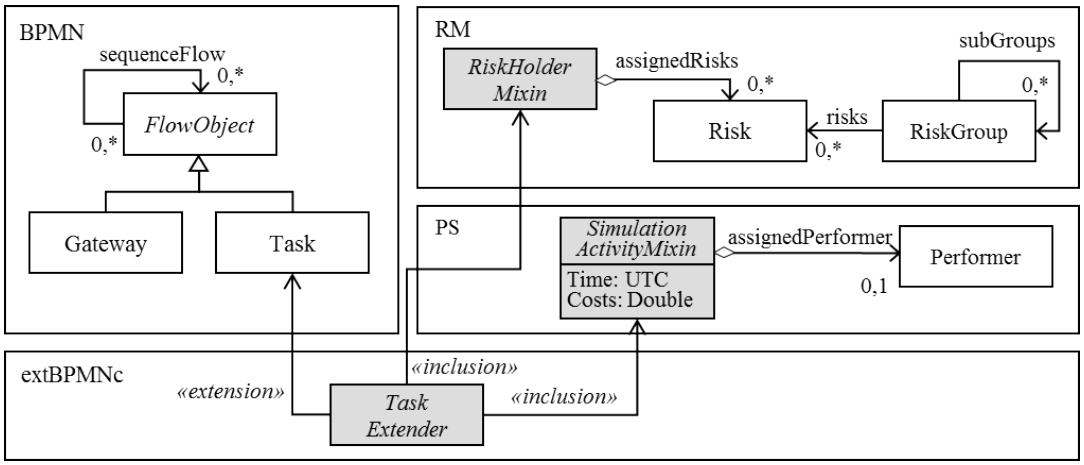


Figure 2: Metamodel customisation based on mixin inclusion and extension composition operators.

tracting the inheritance hierarchy. On the other side, it enables the definition of self-contained aspectual modules which can be flexibly combined, thus fostering reuse and clear separation of concerns. In our example, the class *DerivedTask* may now include mixin classes *RiskHolder* and *SimulationActivity*, in order to obtain their features.

3.3 Extension-based Metamodel Composition

While mixins solve the problem of inheritance singleness and complex inheritance hierarchies, mixins do not tackle the issue of the subclass imperative, when it comes to extending an already existing base metamodel. We still need to create a derived type, when extending the very base element (in our example, class *Task*). In order to address this issue, we introduce the concept of extension and extender-like elements. The extenders may be thought of as inverse mixins. They allow for extensively injecting the features into a parent element without creating a derived element. The extension mimics the semantics of the inheritance, however without a need for a derived type.

The extension-based metamodel composition picks up on the initial idea of invasive software composition (Aßmann, 2003), in which program fragments such as methods and properties may invasively be injected into existing program code by operating on their implicit interfaces using transformative techniques. However, instead of intrusively changing the code, we introduce a native metamodelling language operator that allows to add features to an existing element without the need to modify it. Applied on metamodels, *implicit interfaces* allow for controlled variation points where metamodel elements may be extended. An implicit interface represents an extension point of a metamodel element, which is implicitly defined by the inherent semantics of the underlying metamodelling language. Each element type may have different implicit interfaces. For example, implicit interfaces of the metaclass *Class* are its structural feature sets. Using the extension operator, another element may access such an implicit interface and extend that particular class by injecting e.g. additional member attributes or references. Implicit interfaces are crucial in metamodel customisation, where extensional composition should take place on previously not explicitly defined extension points, or on non-modifiable elements. To combine elements based on extension, a *base element*, an *extender element* and an *extension operator* are required.

- *Base element*. A base element may be any compound metamodel element, for which at least one implicit interface exists. For example, it doesn't make sense to extend elements such as attribute types that do not aggregate other elements and features. In our case, the base element is the class *Task*.
- *Extender element*. An extender element is a compound element, that holds extensions that should be injected to the base element. Since it is a pure *supporting metamodelling construct*, it is a non-instantiable, abstract element. In addition, the extender element must be of the same type as the base element. This is required to implicitly constrain only extensions that are possible for that specific element type.
- *Extension composition operator*. Extension operator is a relation that takes a base element and an extender element as input and extends the base element by injecting extensions based on well-defined implicit interfaces. Like in inheritance,

but inversely, the structural features of the extender element are propagated to the base element without any syntactic modification of the base element. An extender element may extend many base elements. In turn, a base element may be extended by arbitrary extender elements. Hence, the extension operator diminishes the necessity of the subclassing imperative, since base elements may be extended via direct feature injection.

Figure 2 illustrates the revisited customisation of the BPMN module now using the mixin inclusion and extension operators. The modules *RM* and *PS* become self-contained, reusable "mixin" modules, having classes *RiskHolderMixin* and *SimulationActivityMixin* as their central mixin elements. Instead of having the explicit derived class *DerivedTask*, the extension module *extBPMNc* defines the extender class *TaskExtender*, which, on the one side, includes the mixin classes, and, on the other side, extends the class *Task* by injecting its structural features, that of mixins. Mixin and extension operators, when used in combination, allow for flexible, non-intrusive, and modular metamodel customisation by composition. While mixins can be used to define self-contained, reusable modules applicable for arbitrary metamodels, extenders play the role of the *composition glue logic*, i.e. they allow for injecting mixins into existing metamodel fragments.

4 APPLICATION IN ADOxx

This section elaborates on the application of the introduced metamodel composition concepts within the metamodelling language of the metamodelling tool ADOxx (Junginger et al., 2000; Kühn, 2010; OMiLab, 2015). Even though the concepts introduced in the following are defined considering the characteristics of ADOxx metamodelling language, we believe that ideas presented may be translated to other metamodelling languages due to comparable metamodelling expressiveness (see Section 1, Table 1).

4.1 ADOxx Metamodelling Language

ADOxx Meta^2-Model is the meta-metamodel of ADOxx. In the following, we explain its main concepts that will serve as a basis for the further discussion regarding the introduced compositional extensions. Figure 3 illustrates the ADOxx meta-metamodel. In ADOxx, all metamodel constructs are identified by IDs and names. This is represented by the top-level abstract metaclass *AObject* and its subclass *ANamedObject*. Further, various meta-constructs in ADOxx may have attributes. The meta-class *AttributeDefinition* represent an attribute construct that may have a default value, a set of constraints and may be of some simple or complex attribute type. Attributable constructs are generalised by an abstract metaclass *AObjectWithAttributes*. A *ALibrary* is an attributable construct which consists of model types and implicitly of other constructs such as classes and relations. As such, a library represents a bundle of different diagram types. To define diagram types, the concept of model type is used. A *AModelType* is an attributable construct that typifies models and consists of classes and relations. In addition, a model type may have *AModes*, which further subset a model type with respect to available classes and relations. A *AClassDefinition* is an abstract metaclass that can hold attributes, and can be contained by model types. Class definitions may be classes or relations. A *AClass* is the central metamodelling construct used to specify entities of a modelling language. ADOxx supports single inheritance for classes. The construct *ARelationClass* connects classes and/or model types. A relation class connects to other elements indirectly using the concept of endpoint definition. An *AEndpointDefinition* allows classes and model types to be target types of a relation. The number of endpoints defines the arity of the relation, however ADOxx restricts relations to be binary. To be directed, a relation must have at least one endpoint of type From and one of type To. Hence, an endpoint specifies which elements may participate in the relation and how (multiplicity). Furthermore, ADOxx features an additional reuse mechanism by aggregation to increase the support for *intra-level reuse*. Reuse by aggregation is a Cartesian product aggregation function, such that any allowed child element may be aggregated by any allowed parent element. For example, a globally defined attribute definition may be reused by any attributable element and vice versa. Similarly, classes and relations may be reused by model types and modes, endpoints by relations etc. Finally, the central modularisation construct for encapsulating metamodel elements into reusable, modular units is a *AFragment*. A fragment may contain owned or imported elements. Owned elements are existential members of that fragment. Imported elements are those referenced from other fragments. Imported elements contribute to inter-fragment reuse and provide a basis for the application of arbitrary composition operators.

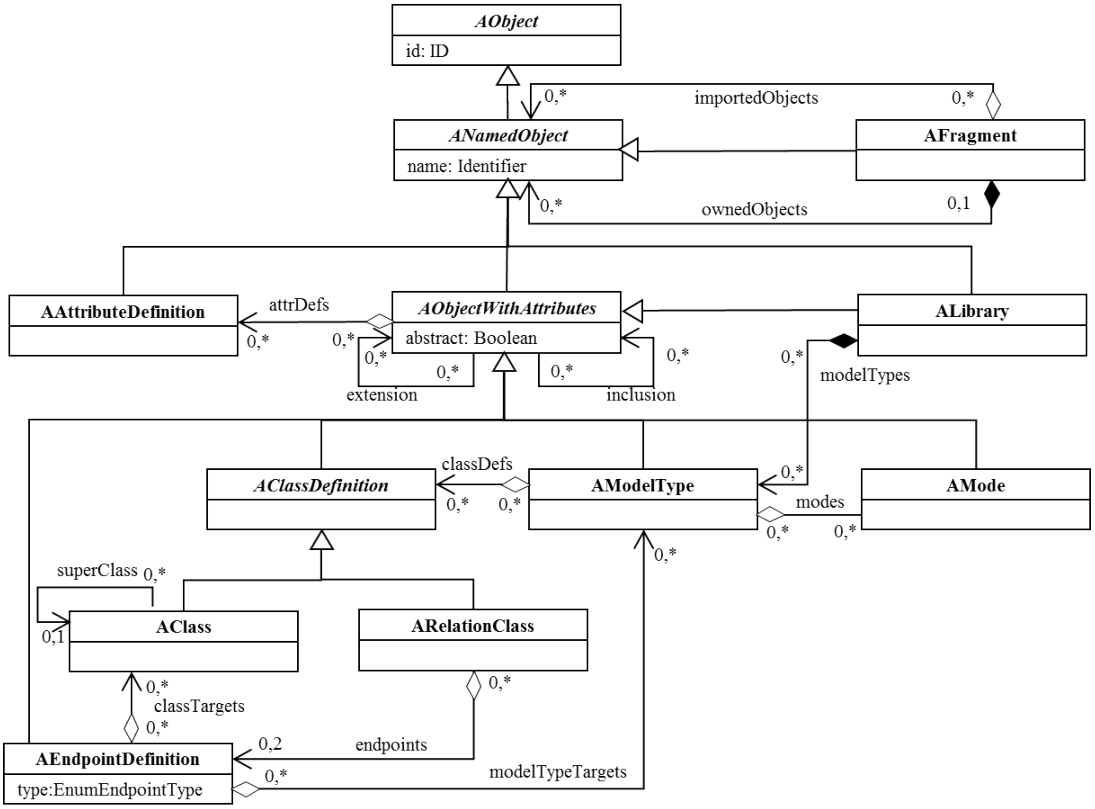


Figure 3: Meta-metamodel of ADOxx featuring Mixins and Extensions.

4.2 Mixins and Extenders in ADOxx

While introducing the idea of mixins and extenders for metamodelling in Section 3, we also defined on which type of metamodelling constructs mixin inclusion and extension composition operators can be applied. In particular, we defined what the parent element, the base element, as well as, what the mixin and the extender elements may be. For all elements applies that they must be of the compound type, i.e. that they must contain an extensible set of child elements. Furthermore, mixin and extender elements must be abstract elements. Therefore, in ADOxx we define both mixin inclusion and extension operator as relationships on the most general level of attributable constructs, i.e. at the metaclass *AObjectWithAttributes* (see Figure 3). By saying that an attributable construct may mixin and/or extend another attributable construct, we allow for the application of these operators for all compound subtypes such as the class, the model type, the relation class, the endpoint, and the library. This is desirable as all of the compound elements have at least attributes as child elements and, in addition, other contained elements, too. For example, an endpoint has attributes, and a set of target classes or target model types. However, there is a number of constraints that need to be obeyed. In the following, we mention the most important ones:

- **Constraint 1:** *Same type mixin composition*. Mixin inclusion relationship can only connect elements of the same metatype. For example, a class can mix in another class, but cannot mix in a model type.
- **Constraint 2:** *Same type extension composition*. Extension relationship can only connect elements of the same metatype. For example, a model type can extend another model type, but cannot extend a class.
- **Constraint 3:** *Abstract mixin element*. The target element of the mixin inclusion relationship must be declared as abstract.
- **Constraint 4:** *Abstract extender element*. The source element of the extension relationship must be declared as abstract.
- **Constraint 5:** *Acyclic mixin dependency*. The mixin element cannot mix in itself, neither directly nor indirectly.
- **Constraint 6:** *Acyclic extender dependency*. The extender element cannot extend itself, neither directly nor indirectly.

- **Constraint 7:** *Acyclic mixin/extender dependency*. A parent element cannot mix in a mixin element, if it at the same time extends it, neither directly nor indirectly.

The semantics of both operators are common for each instantiable compound metaclass (class, relation class, endpoint definition, model type, mode, library), with regard to inclusion and extension of attributes. Moreover, the semantics of the inclusion operator is very similar to the inheritance of attributes, whereas for extension, it acts as a kind of inverse inheritance of attributes. Hence, in ADOxx, we implement the semantics for mixin inclusion and extension relationship based on the following definitions. First, we introduce the common semantics for all metaclasses that are attributable elements (all subclasses of the metaclass *AObjectWithAttributes*).

- **Definition 1:** *Inclusion of Objects with Attributes*. Given the parent element Ep with the set of attributes Sp, and the mixin element Em with a set of attributes Sm, Ep *includes* Em by aggregating all attributes of Sm into Sp.
- **Definition 2:** *Extension of Objects with Attributes*. Given the base element Eb with the set of attributes Sb, and the extender element Ee with a set of attributes Se, Ee *extends* Eb by aggregating all attributes of Se into Sb.

However, since each metaclass (subclass of objects with attributes) has a different compound structure, i.e. the set of containable structural features on which the operators are applied, the semantics of operators vary for each such metaclass. For example, model type mixin inclusion aggregates all classes of a model type to the parent model type. Inversely, a model type extender inserts all its class members into the base model type. Since listing of all additional definitions for each construct would exceed the limits of the underlying work, we focus only on those elements which, as we will see, we also use in our running example - classes and endpoint definitions. While for classes no further structural containment exists, for the endpoint we define mixin inclusion and extension as follows:

- **Definition 3:** *Inclusion of Endpoint Definitions*. Given the parent endpoint EPp with the set of target classes Sc_1 and the set of target model types Sm_1, and the endpoint mixin EPm with a set of target classes Sc_2 and the set of target model types Sm_2, EPp *includes* EPm by aggregating all target classes of Sc_2 into Sc_1 and all target model types of Sm_2 into Sm_1.
- **Definition 4:** *Extension of Endpoint Definitions*. Given the base endpoint EPb with the set of target classes Sc_1 and the set of target model types Sm_1, and the endpoint extender EPe with the set of target classes Sc_2 and the set of target model types Sm_2, EPe *extends* EPb by aggregating all target classes of Sc_2 into Sc_1 and all target model types of Sm_2 into Sm_1.

Finally, both inclusion and extension relationships are applied transitively.

4.3 Applying Mixins and Extenders

In the following, we revisit the running example from Section 2 and the conceptual solution from Section 3, in order to exemplify the application of mixins and extenders in ADOxx. Figure 4 illustrates the revisited solution. We now apply ADOxx metaclasses to implement metamodel fragments *BPMN*, *RM*, *PS* and *extBPMNc*. In doing so, we use UML stereotypes to denote corresponding metaclasses from ADOxx. Note that we explicitly model cross-package relationships, instead of re-modelling the imported classes, in order to save the space in the diagram. We define the abstract class *TaskExtender*, which, on the one side, includes the mixin class *SimulationActivityMixin*, and on the other side, extends the class *Task* by inserting the structural features (that of the included mixin). Note that in ADOxx, only attributes *Time* and *Costs* will be propagated as the only member features of the class *SimulationActivityMixin*. Since relations between classes in ADOxx are defined over endpoints as first-order constructs, a relation of a class is syntactically not an inherent feature of that class. Instead, a class is a target, a structural feature of a relation endpoint. Hence, to extend the class *Task* with the relation *AssignedPerformer*, we define the endpoint extender *FromAPExtender*, which, on the one side, targets the class *Task*, and on the other side, extends the corresponding endpoint *FromAP* of the relation *AssignedPerformer*. Similarly, we define the endpoint extender *FromARExtender* for the endpoint *FromAR* of the relation *AssignedRisks*, in order to allow for tasks to connect to performers. As a result of composition, the class *Task* contains both risk-related and simulation features.

4.4 Application Evaluation

One may argue that introduced concepts such as mixins, extenders and, in general, modular thinking, although powerful, bring an additional level of complexity in metamodelling. While this argument is true for a one-time metamodel customisation project, the true benefit of modular customisation with mixins and extenders becomes visible with repeated use. In order

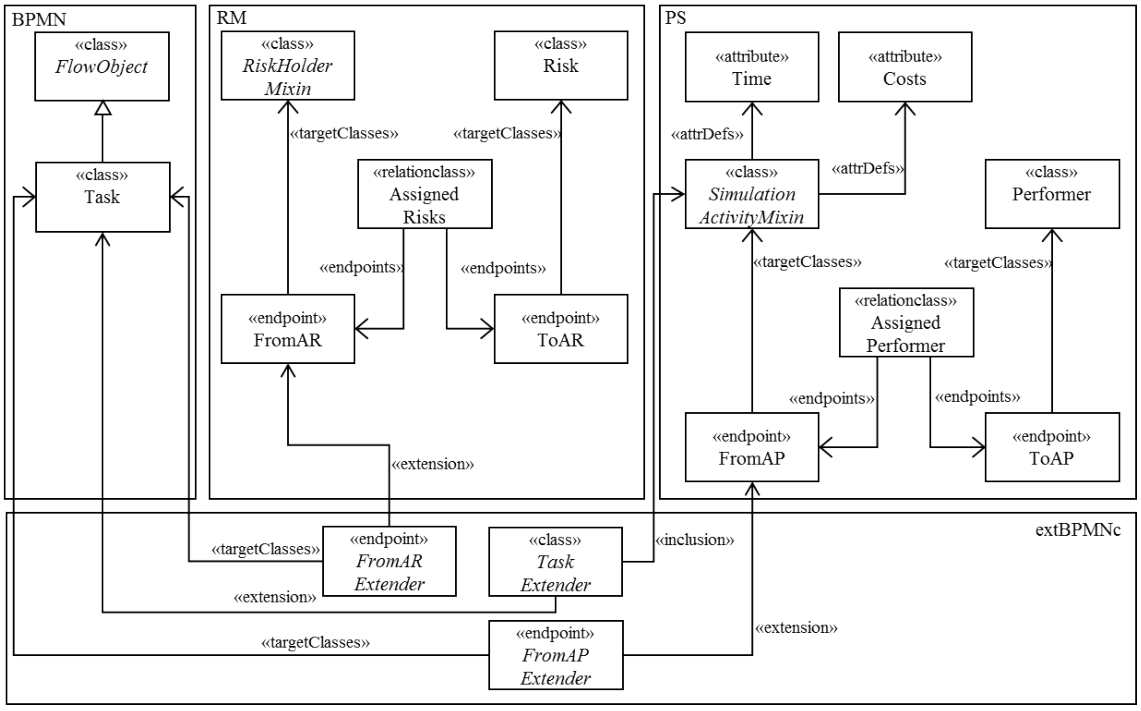


Figure 4: Application of mixin inclusion and extension composition operators in ADOxx.

Table 2: Comparison of customisation effort with basic metamodelling constructs and with mixins and extenders for modular metamodel customisation (estimation in story points (SP)).

Customisation effort	Basic ADOxx constructs	Mixins and Extenders
Initial customisation	3 SP	3.5 SP
Migration of customisation	2 SP	0.5 SP
Reuse of customisation	2 SP	0.5 SP

to evaluate this thesis, we conducted a survey among experienced ADOxx language and metamodel engineers. As part of the survey, we asked engineers to provide two effort estimations for the sample metamodel customisation project from the introduced running example. Given the introduction of the new modular customisation concepts in ADOxx, they provided effort estimations to conduct the work 1) based on basic metamodelling constructs, 2) based on new modular metamodel customisation constructs. The following customising tasks have been considered: 1) initial (from scratch) customisation, 2) migration of the customisation to a new base metamodel version, 3) reuse of customisation for another metamodel. The results are summarised in the Table 2.

As expected, the initial customisation effort estimation in average was slightly higher when using new constructs (mixins and extenders). This was explained by the fact, that this kind of customisation required upfront design in modules for future reuse. However, the estimated effort for the migration was significantly lower when using mixins and extenders. This was mainly due to the fact that the extension could be ported as a mixin module to the new version and injected via extenders without needing to migrate data (refer to the subclassing imperative issue of inheritance). Regarding the reuse of customisation in another project, the effort was clearly lower, since the mixin extension module could be reused as-is, with the only effort of defining new extender class.

5 RELATED WORK

In the area of programming languages, the idea of mixins has been around for years. The term was coined in the language Flavors (Moon, 1986), however, mixins have been initially defined as a formal language construct for language CLOS (Bracha and Cook, 1990). Mixins found usage in OOPLs such as Smalltalk (Bracha and Griswold, 1996), and Scala (Odersky et al., 2004). GPLs such as C++, that do not support mixins natively, aim at emulating the behaviour of mixins based on parameterised inheritance and template classes (Smaragdakis and Batory, 2001). Similarly, in (Ancona et al., 2000) an exten-

sion for Java has been proposed called *Jam*, to allow for mixin-based class composition. As for extensions, the initial idea of an *Extend* operator that injects program code fragments into an existing program code at implicit interfaces was proposed in (Aßmann, 2003) as a part of the broader approach of invasive software composition. Similar approach to extend existing types exists in C#, in which so-called *Extension Methods* allow for injecting methods into an existing base type without the need to create a new derived type, recompile, or otherwise modify the base type (MSDN, 2015).

In modelling language engineering, several techniques for metamodel and DSML composition have been proposed (Vallecillo, 2010). We focus on those that allow for metamodel customisation beyond inheritance. UML 2 provides the profile mechanism for metamodel customising, particularly applicable for UML 2 family of languages. In *UML Profiles*, selected concepts of the UML metamodel may be extended using stereotypes. With UML 2, profiles have been improved from a lightweight customisation approach to a sound mechanism for both metamodel customisation and for the design of new languages (Selic, 2007). In the current UML version 2.4.1 (OMG, 2011), the concept of the *Stereotype* is a full metaclass that specialises the *Class* concept and extends it through the explicit association *Extension*. As a subclass of the *Class*, a stereotype may own properties that extend the base class. Furthermore, stereotypes allow for the creation of new associations between stereotypes and other metamodel elements. The notion of a stereotype is comparable to our extender element, and the extension association with our extension operator. However, instead of introducing a separate metaclass for it, we add extensibility capability to the corresponding metaclass itself, with the only constraint that such extender element must be abstract. Hence, our extender element is simply a supporting metamodelling construct that extends an existing element while not being instantiable on the model-level. Since the extension occurs in design and compile time, no further model-level mechanisms are required to correlate the instances of a stereotype with instances of a base element. Although, it is claimed that profiles are made generic and compatible for any MOF-based language (Selic, 2007; OMG, 2011), to our best knowledge, we are not aware of any other profile applications than those for UML.

In (Langer et al., 2012), the idea of profiles is applied on Ecore, however, not on the meta-metamodel level but on the metamodel level through metalevel lifting. The so-called *EMF Profiles* help to customise arbitrary DSMLs that are based on EMF Ecore. Since the same idea of UML Profiles is applied, similarities and differences to our approach apply as mentioned before for UML profiles. Furthermore, two additional mechanisms are added that increase profile reuse, *Generic Profiles* and *Meta Profiles*. Generic profiles are based on generic types. This is inherently supported in our approach through the usage of abstract elements when defining mixin and extender modules. Meta Profiles aim at applying extensions to the constructs of the meta-metalevel, such that extensions are applicable for all DSMLs. This is an interesting approach for massive extensions, we do not yet support.

In (Braun and Esswein, 2015) a similar idea of adapting the UML Stereotype concept towards a mechanism for generic metamodel extensions, however only on the conceptual level, is proposed. Unlike in (Langer et al., 2012) and similar to our approach, the authors, propose an extension on the meta-metamodel level, with an application focus on enterprise modelling languages. As in the UML/MOF stereotype mechanism, the stereotype construct is defined as a separate metaclass, however not only for the Class construct, but multiplied for each metaclass type (model type, property etc.). Each stereotype construct has a limited set of extension possibilities. In our approach, we fully reuse existing metaclasses as abstract constructs to capture extensions, and, instead, define precise extension semantics on the extension operator.

Another metamodel extensibility concept common to UML and MOF is the *PackageMerge*. Package merge is used to merge elements and the content of two packages. As noted in (Selic, 2011), package merge allows for incremental metamodel refinement, because one can define an extending element (increment) with the same name as the base element, add additional properties to it and merge it with the base. The semantics of the package merge is similar to that of generalisation, with a difference that the derived element has the same name as the base. In comparison to our work, package merge has similarities to both mixin and extension. However, the difference is that we do not rely on name matching algorithm but on explicit relationships defined by language engineer, which contributes to an increased soundness in metamodel composition. Furthermore, both mixins and extensions are applied on the level of metamodel elements to allow for a fine-grained and precise extension definition, whereas package merge operates on the level of packages, which potentially opens the door for uncontrolled reuse and the need for metamodel pruning techniques such as *Package Un-*

merge (Fondement, Frédéric and Muller, Pierre-Alain and Thiry, Laurent and Wittmann, Brice and Forestier, Germain, 2013). Another MOF extensibility capability is the *Extension*, a lightweight approach to annotate existing metamodel elements with *Tags* that, however, solely represent simple name-value pairs.

Apart from the mainstream metamodelling approaches, in (de Lara and Guerra, 2013), generic programming techniques such as *concepts*, *templates* and *mixin layers* are applied for metamodelling in order to increase the support for abstraction, modularity, reusability and extensibility of (meta)models and corresponding model management operations. Focusing on the usage of mixins, mixin layers rely on templating technique that allows for defining templated metamodel extensions (mixin layers), that can be applied on metamodels that conform to template parameters. The basic idea is to use the parameterised inheritance to realise a generic mixin. The "instantiation" of the template binds the mixin layer to a concrete metamodel that is the subject to extension and that conforms to the structure defined by the parameter type (concept). While the authors introduce templates and template instantiation to realise generic mixins and their application, we rely on the abstract metaclasses, and mixin and extension operators. Instead of applying the parameterised inheritance to achieve the flexibility of mixin applications, we define extenders that insert mixins into arbitrary elements. However, we believe that the two approaches are complementary. While acknowledging the power of templates for specifying the generic concepts that promote abstraction, modularity and reuse, our mixin inclusion, and, in particular, *extension* operator may be applied on the level of templates to allow for mixing and extending of template definitions themselves.

Finally, in (Jézéquel et al., 2013) an approach to design domain-specific languages based on multiple meta-languages within the Kermeta language workbench is proposed. The composition operators *aspect* and *require* are introduced that allow for the composition of language elements such as abstract syntax, static semantics, and behavioral semantics. Inspired from the open class concept, the aspect allows to reopen a previously created class and to add features. In what it does, the aspect is similar to our extension operator. Differently to our approach, the focus is on modular composition of language concerns, whereas we concentrate on the pivotal language aspect, the metamodel. The authors state that the modularisation does not apply for the metamodel aspect (for which definition the EMOF is used).

6 CONCLUSIONS

This work represents a contribution to the field of metamodel composition and customisation, in the context of metamodel-based modelling language engineering. Our approach is based on the notions of Mixins and Extenders and appropriate composition operators, that allow for flexible, modular metamodel customisation. Mixin and extension-based metamodel composition facilitate reuse and contribute to more flexibility and overall efficiency in metamodel definition. Mixins allow for the creation of reusable aspectual metamodel element extensions that can be combined by arbitrary metamodel elements using the mixin inclusion operator. Furthermore, the extension operator allows for the injection of structural features into otherwise non-modifiable base metamodel elements in a non-intrusive way by relying on their implicit interfaces. While mixin inclusion complements single inheritance, and, at the same time, represents a lightweight alternative to multiple inheritance, extension resolves the issue of subclass imperative, an important issue in metamodel customisation. We illustrated the usefulness of the approach based on a running example of the simplified BPMN metamodel customisation. We also elaborated on the application of our approach within the metamodelling tool ADOxx. Although we explained the syntax and semantics of the operators based on ADOxx, the concepts may be mapped to other metamodelling languages and tools, as well.

In this work, we focused primarily on the metamodel as a pivotal part of language definition. A part of our future work at OMILab [1] will focus on investigating how modular approach based on mixins and extensions may be applied on the composition of other language elements such notation and semantics. Furthermore, with an increased usage of mixins and extenders in metamodel composition, we will work on identifying common patterns for modular metamodel engineering.

REFERENCES

Ancona, D., Lagorio, G., and Zucca, E. (2000). Jam- a Smooth Extension of Java with Mixins. In *ECOOP 2000ŮObject-Oriented Programming*, pages 154–178. Springer.

Aßmann, U. (2003). *Invasive Software Composition*. Springer.

Bracha, G. (1992). *The Programming Language Jigsaw:*

[1] http://omilab.org

Mixins, Modularity and Multiple Inheritance. PhD thesis, The University of Utah.

Bracha, G. and Cook, W. (1990). Mixin-based inheritance. In *ACM SIGPLAN Notices*, volume 25, pages 303–311. ACM.

Bracha, G. and Griswold, D. (1996). Extending Smalltalk with Mixins. In *Workshop on Extending Smalltalk*.

Braun, R. and Esswein, W. (2015). Designing Dialects of Enterprise Modeling Languages with the Profiling Technique. In *19th IEEE International Enterprise Distributed Object Computing Conference, EDOC 2015, Adelaide, Australia, September 21-25, 2015*, pages 60–67.

de Lara, J. and Guerra, E. (2013). From Types to Type Requirements: Genericity for Model-Driven Engineering. *Software & Systems Modeling*, 12(3):453–474.

Fondement, Frédéric and Muller, Pierre-Alain and Thiry, Laurent and Wittmann, Brice and Forestier, Germain (2013). Big Metamodels are Evil. In *Model-Driven Engineering Languages and Systems*, pages 138–153. Springer.

Herbst, J., Junginger, S., and Kühn, H. (1997). Simulation in Financial Services with the Business Process Management System ADONIS. In *Proceedings of the 9th European Simulation Symposium*.

Jézéquel, J.-M., Combemale, B., Barais, O., Monperrus, M., and Fouquet, F. (2013). Mashup of Metalanguages and its Implementation in the Kermeta Language Workbench. *Software & Systems Modeling*, pages 1–16.

Junginger, S., Kühn, H., Strobl, R., and Karagiannis, D. (2000). Ein Geschäftsprozessmanagement-Werkzeug der nächsten Generation - ADONIS: Konzeption und Anwendungen. *Wirtschaftsinformatik*, 42(5):392–401.

Kelly, S., Lyytinen, K., and Rossi, M. (1996). Metaedit+ a Fully Configurable Multi-User and Multi-Tool CASE and CAME Environment. In *Advanced Information Systems Engineering*, pages 1–21. Springer.

Kühn, H. (2010). The ADOxx Metamodelling Platform. In *Workshop on Methods as Plug-Ins for Meta-Modelling, Klagenfurt, Austria*.

Langer, P., Wieland, K., Wimmer, M., Cabot, J., et al. (2012). EMF Profiles: A Lightweight Extension Approach for EMF Models. *Journal of Object Technology*, 11(1):1–29.

Ledeczi, A., Maroti, M., Bakay, A., Karsai, G., Garrett, J., Thomason, C., Nordstrom, G., Sprinkle, J., and Volgyesi, P. (2001). The Generic Modeling Environment. In *Workshop on Intelligent Signal Processing, Budapest, Hungary*, volume 17.

Moon, D. A. (1986). Object-oriented programming with Flavors. In *ACM Sigplan Notices*, volume 21, pages 1–8. ACM.

MSDN (2015). Extension Methods (C# Programming Guide). https://msdn.microsoft.com/en-us//library/bb383977.aspx.

Odersky, M., Altherr, P., Cremet, V., Emir, B., Maneth, S., Micheloud, S., Mihaylov, N., Schinz, M., Stenman, E., and Zenger, M. (2004). An overview of the Scala programming language. Technical report, EPFL.

OMG (2011). UML 2.4.1 Infrastructure Specification. http://www.omg.org/spec/UML/2.4.1/Infrastructure/PDF/.

OMG (2013). Business Process Model and Notation (BPMN) Version 2.0.2. http://www.omg.org/spec/BPMN/2.0.2/PDF.

OMG (2014). Meta Object Facility (MOF) Version 2.4.2. http://www.omg.org/spec/MOF/2.4.2/.

OMILab (2015). ADOxx Metamodelling Platform. http://www.adoxx.org.

Rausch, T., Kuehn, H., Murzek, M., and Brennan, T. (2011). Making BPMN 2.0 Fit for Full Business Use. *BPMN 2.0 Handbook Second Edition*, page 189.

Selic, B. (2007). A Systematic Approach to Domain-specific Language Design using UML. In *Object and Component-Oriented Real-Time Distributed Computing, 2007. 10th IEEE International Symposium on*, pages 2–9. IEEE.

Selic, B. (2011). The Theory and Practice of Modeling Language Design for Model-Based Software Engineering - A Personal Perspective. In *Generative and Transformational Techniques in Software Engineering III*, pages 290–321. Springer.

Smaragdakis, Y. and Batory, D. (2001). Mixin-based Programming in C++. In *Generative and Component-based Software Engineering*, pages 164–178. Springer.

Steinberg, D., Budinsky, F., Merks, E., and Paternostro, M. (2008). *EMF: Eclipse Modeling Framework*. Pearson Education.

Vallecillo, A. (2010). On the Combination of Domain Specific Modeling Languages. In *Proceedings of European Conference on Modelling Foundations and Applications, 2010. (ECMFA 2010)*, volume 6138 of *LNCS*, pages 305–320. Springer.

Especifying the Enterprise and Information Viewpoints for a Corporate Spatial Data Infrastructure using ICA's Formal Model

Italo L. Oliveira[1], Jugurta Lisboa-Filho[1], Carlos A. Moura[2] and Alexander G. Silva[2]
[1]*Department of Informatics, Federal University of Viçosa, Viçosa, Minas Gerais, Brazil*
[2]*Companhia Energética de Minas Gerais (CEMIG), Belo Horizonte, MG, Brazil*
{italo.oliveira, jugurta}@ufv.br, {camoura, ags}@cemig.com.br

Keywords: Spatial Data Infrastructure, RM-ODP, Enterprise Viewpoint, Information Viewpoint.

Abstract: The International Cartographic Association (ICA) has proposed a formal model to describe Spatial Data Infrastructure (SDI) using three of the five viewpoints of the RM-ODP (Reference Model for Open Distributed Processing) framework, which was later adapted by other researchers. However, the adapted ICA model has not been validated for corporate-level SDI. The *Companhia Energética de Minas Gerais* (Minas Gerais Power Company - Cemig) seeks to develop an SDI to aid in discovering and reutilizing spatial data within and outside the corporation. The present study aimed to assess the use of the model proposed by the ICA to specify corporate-level SDI using SDI-Cemig as a case study by describing the viewpoints Enterprise and Information. These viewpoints from the adapted ICA model have proved appropriate to describe SDI-Cemig, whose differences are due to the SDI's peculiarities. Although a single study cannot validate the ICA model for a whole SDI level, this research shows that the adapted ICA model can be used to describe the viewpoints Enterprise and Information in corporate SDI.

1 INTRODUCTION

Geospatial data are those referenced in relation to the ground surface and are essential to aid in an organization's decision-making and planning. However, according to Nebert (2004) and Rajabifard and Williamson (2001), geospatial data are a costly resource both in time and money involved in surveying them. In order to cut down the costs associated with using and obtaining geospatial data, the Spatial Data Infrastructure (SDI) concept was created.

There are several definitions for SDI. Rajabifard and Williamson (2001) define SDI as an environment in which the users reach their goals by using technologies and collaboration. Harvey et al. (2012) consider the SDI a concept that aids in sharing data and geospatial services among different users of a given community.

In order to help share and discover geospatial data and services, the SDIs are organized hierarchically. Figure 1 presents the SDI hierarchy and the nomenclatures used in the present study.

According to Hjelmager et al. (2008), the SDI concept is very broad and leads to different forms of development both at the organizational and technical level, as pointed out by Cooper et al. (2013). Thus, the International Cartographic Association (ICA) has developed a model to describe SDI regardless of the technologies or implementations (Hjelmager et al., 2008), a concept that was later extended by Cooper et al. (2011); Béjar et al. (2012); Cooper et al. (2013); and Oliveira and Lisboa-Filho (2015).

However, the use of ICA's formal model for SDI has not been evaluated to develop corporate-level SDI yet. The *Companhia Energética de Minas Gerais* (Cemig) is a mixed-economy company acting in the electricity sector composed of over 200 partners and controlled by the government of the

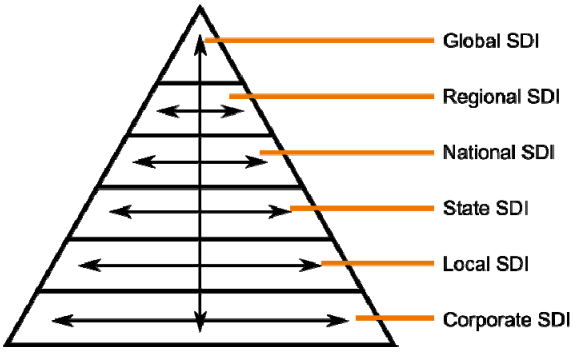


Figure 1: SDI hierarchy – Adapted from Rajabifard and Williamson (2001) and Crompvoet (2001).

Oliveira, I., Lisboa-Filho, J., Moura, C. and Silva, A.
Especifying the Enterprise and Information Viewpoints for a Corporate Spatial Data Infrastructure using ICA's Formal Model.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 271-282
ISBN: 978-989-758-187-8

state of Minas Gerais (Brazil). Cemig seeks to develop an SDI, named SDI-Cemig, to standardize the processes that use the company's geospatial data, thus helping such data be shared and surveyed.

The present study presents the use of ICA's formal SDI model under SDI-Cemig's specification while detailing the viewpoints Enterprise and Information and verifying whether this model allows a corporate SDI to be appropriately described.

The remaining of the paper is structured as follows. Section 2 describes ICA's formal SDI model, detailing the viewpoints Enterprise and Information of an SDI. Section 3 presents the specification of the viewpoints Enterprise and Information for SDI-Cemig. Section 4 discusses the results found in the present research, while Section 5 presents some final considerations of the study.

2 ICA'S FORMAL MODEL

According to Hjelmager et al. (2008), ICA's formal SDI model (henceforth called only formal model) is a model that describes SDI regardless of technologies, policies, or implementations. In order to describe SDI, the ICA chose to use the RM-ODP (Reference Model for Open Distributed Processing) framework.

RM-ODP is an architectural framework standardized by the International Organization for Standardization/International Electrotechnical Commission (ISO/IEC) that is able to describe heterogeneous distributed processing systems by using viewpoints (Farooqui et al., 1995).

According to Raymond (1995), the use of the viewpoint concept allows describing complex distributed systems as smaller models, each of which describes different relevant issues to different users of the system. RM-ODP uses the following viewpoints: Enterprise, Information, Computation, Engineering, and Technology. Figure 2 presents the five viewpoints and the relationship among them.

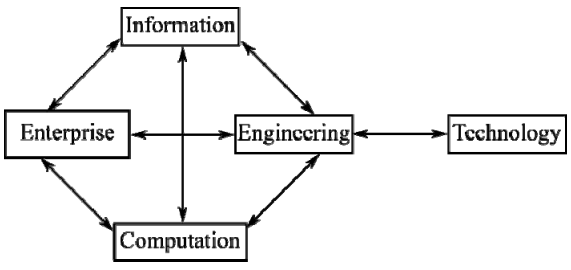


Figure 2: RM-ODP framework viewpoints – Adapted from Hjelmager et al. (2008).

The viewpoint Enterprise describes the system's policies, scope, goal, and requirements for the organization. The viewpoint Information details the data semantics and the behavior in the system, whose behavior will be restricted/determined by the policies defined in the viewpoint Enterprise (Farooqui, Logrippo and de Meer, 1995) (Hjelmager et al., 2008). According to Cooper et al. (2013), the viewpoint Computation describes the components that make up the system and their interactions through the interface with no concern about the components' physical distribution. The viewpoint Engineering, according to Farooqui, Logrippo and de Meer (1995), "identifies the requirements and features needed for the system to support the model described in the viewpoint Computation." Finally, the viewpoint Technology details the technological devices used by the system.

ICA's formal model describes only the viewpoints Enterprise, Information, and Computation. According to Hjelmager et al. (2008), the viewpoints Engineering and Technology heavily depend on the implementation and are not considered in ICA's model. The viewpoints Enterprise and Information will be described in the sub-sections below. The viewpoint Computation will not be described since it is not relevant for this study.

2.1 Enterprise Viewpoint

The viewpoint Enterprise, according to Hjelmager et al. (2008), describes the actors and the relation among the different parts of the system.

Figure 3 shows the relationship among the different compounds that make up the SDI through a diagram of UML classes. In the diagram, the SDI is the central compound and its attributes are the scope and a plan for its implementation. An SDI is formed by products, which are in turn formed by geospatial data and services from the SDI. The acquisition and use of these products is the reason why a user will use the SDI. Hence, the *Product* can be considered the core part of the SDI (Hjelmager et al., 2008).

The *Metadata* will describe and be used by the *Product*, and will be managed by the *Processing Tools* to aid in the discovery and use of geospatial data and services. The component *Processing Tools* represents the systems that carry out some sort of geospatial data processing and will connect to the SDI through the component *Connectivity*, which will use a certain *Technology* to perform its role.

According to Hjelmager et al. (2008), the component *Policies* is responsible for defining the

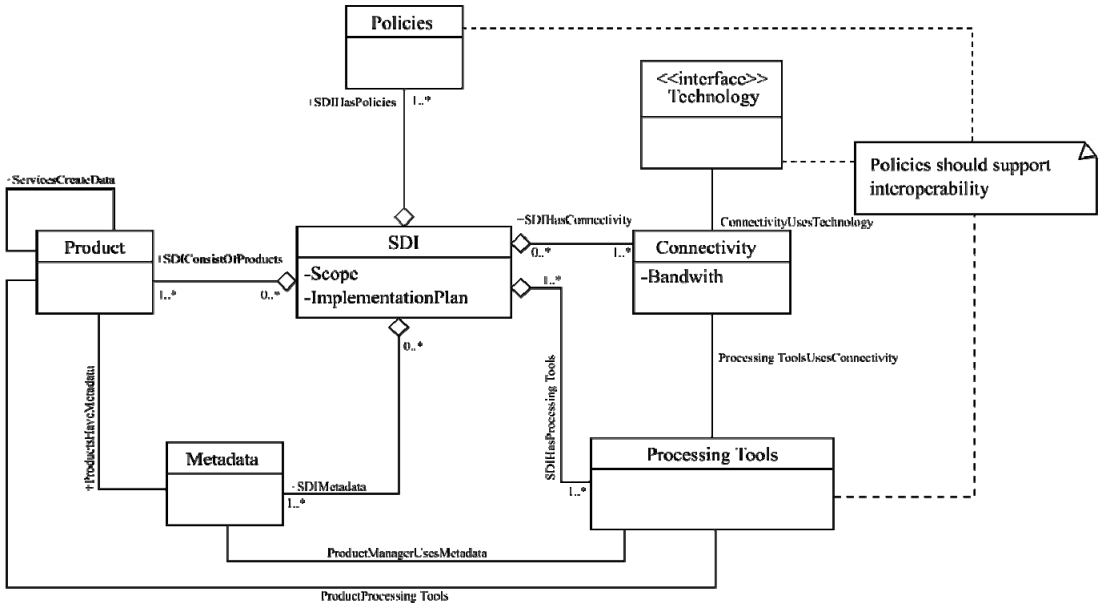


Figure 3: Components that make up an SDI – Hjelmager et al. (2008).

policies that will restrict and determine the SDI's functioning and evolution. Although this component is represented by a single class, the component *Policies* may be specialized into several other classes, which will be shown ahead.

The actors are individuals with a stake on the SDI's success and may use it or contribute to it. Hjelmager et al. (2008) defined five main actors for the SDI, which were expanded by Cooper et al. (2011) and Béjar et al. (2012). However, there are differences in semantics and terminology between the actors by Hjelmager et al. (2008) and Cooper et al. (2001) and those proposed by Béjar et al. (2012). This same characteristic holds true regarding the SDI's policies.

Oliveira and Lisboa-Filho (2015) unified the actors and policies proposed by the ICA with those proposed by Béjar et al. (2012). This way, the designers that may use ICA's model will have a single set of possible actors and policies, which facilitates communication and knowledge exchange among designers.

Figure 4 presents the six main actors an SDI may have: *User*; *Producer*; *Operational Body*; *Governing Body*; *Broker*; *Value-Added Reseller*; and *Provider*.

According to Oliveira and Lisboa-Filho (2015), the *User* is the actor that will use the resources offered by the SDI to reach his or her goals. The *Producer* is responsible for producing the SDI's data and services while the *Provider* makes these data and services available. The *Governing Body* is responsible for the SDI's administration and its attributions include creating, changing, and removing policies. The *Broker*'s role is to aid in the negotiations between providers and users. The *Value-Added Reseller* (VAR) modifies an existing product and makes it available in the SDI as a new product. Finally, the *Operational Body* is responsible for all the technical side of the SDI's functioning. All actors are specialized to describe their attributions in more details. The specializations can be found in Oliveira and Lisboa-Filho (2015).

Table 1 presents the policies unified by Oliveira and Lisboa-Filho (2015). The policies were specialized into: *Business Model*, *Promotion*, *Standards*, *Education*, and *Constraints*. The descriptions and specializations of each type are shown in Table 1.

2.2 Viewpoint Information

According to Hjelmager et al. (2008), the viewpoint Information in the RM-ODP framework describes the system data, from their semantics to their behavior, which are regulated by the policies defined in the viewpoint Enterprise. In the case of an SDI, Hjelmager et al. (2008) consider as data the products offered by the SDI, i.e., the geospatial data and services.

Figure 5 describes the relationship of the products with the other SDI components using the UML class diagram. The class *Product*, for being the most relevant object in the viewpoint Information, is the center of the diagram. The class

Table 1: SDI policies after the unification – Oliveira and Lisboa-Filho (2015).

Policies		Description
Business Model	Governance	Determines the decision-making process
		Regulates the policy-creation process
	Membership	Determines the relationships among the SDI members
	Quality	Defines the quality levels established in the SDI
	Access	Determines how the SDI products can be accessed and who can do it
	Role Assignment	Defines the responsibilities (actor roles) of the SDI users
	Funding	Defines how the resources will be forwarded to develop and maintain the SDI
Promotion	-	How the SDI will be advertised
Standards	-	Defines the standards adopted by the SDI
	Foundation	Defines the main SDI products
Education	-	Determines the trainings the SDI users may take part in
	Best Practices	Practices that must be adopted by the users member of the SDI
Constraints	Legal Constraints	Restrictions imposed by local laws of where the SDI is located
	Business Agreements	Restrictions existing due to contract between companies

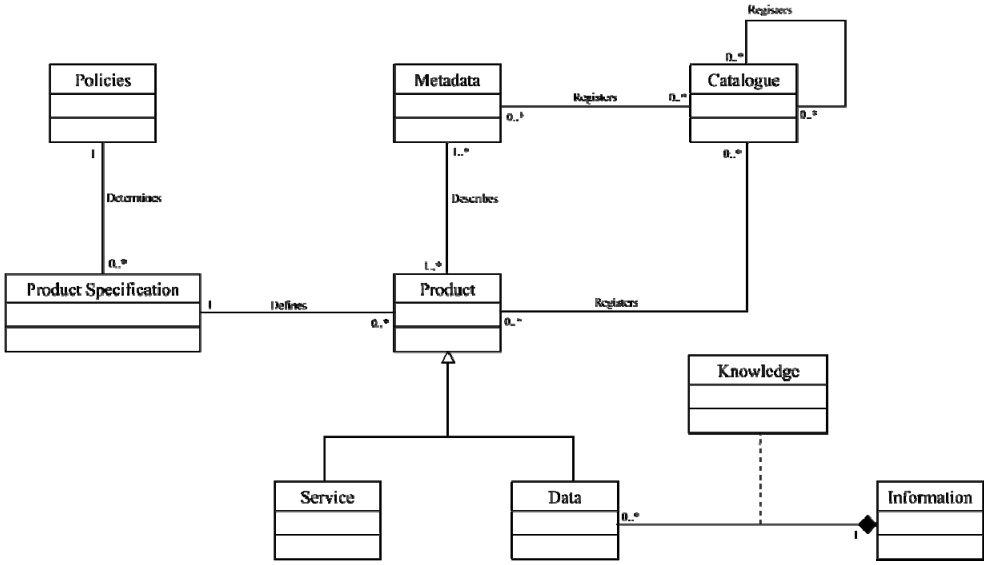


Figure 5: Class diagram for the viewpoint information – Hjelmager et al. (2008).

Policies represents the policies defined in the viewpoint Enterprise, which will restrict and target the product specifications, which are represented by the class *Product Specification* (Hjelmager et al., 2008).

The *Products* are described by the metadata (class *Metadata*) and both are recorded in catalogs (class *Catalog*), which may contain other catalogs to allow for a hierarchy to be created. The products can be classified into ervices and data (either geospatial or not). The data are used, aided by previous knowledge, as a source of information, which may generate new knowledge (Hjelmager et al., 2008).

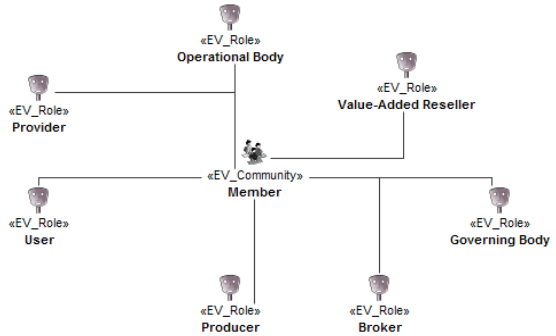


Figure 4: Main actors of an SDI after the unification – Oliveira and Lisboa-Filho (2015).

3 SDI-CEMIG

As specified in Section 1, Cemig seeks to develop an SDI to help share and use geospatial data in the companies that make up the conglomerate. The model adapted from the ICA was used to specify the SDI-Cemig so as to guarantee that the basic SDI concepts in the literature would be contemplated during the specification phase. The sub-sections below describe the viewpoints Enterprise and Information of SDI-Cemig.

3.1 Viewpoint Enterprise

As described in sub-section 2.1, the ICA has described the parts that make up the SDI and the possible actors that may interact with it. The components and actors were identified in SDI-Cemig to check whether they properly describe corporate SDI.

3.1.1 Components of SDI-Cemig

The SDI is considered the central element in Figure 3 and has a scope and implementation plan (Hjelmager et al., 2008). The scope of SDI-Cemig is to make available online a set of geospatial layers considered essential to the companies in the electric sector and that may be used by Cemig's employees and clients, besides offering services to visualize and discover geospatial data. The implementation plan of SDI-Cemig will be publicized by the end of the SDI's development.

The component *Product* is made up of the SDI's geospatial data and services. SDI-Cemig has the data of the geospatial layers considered basic for Cemig, i.e., they are essential layers to the working of the processes that involve geospatial data and are described by the *Foundation* policies and detailed in the conceptual model in sub-section 3.2.1.

SDI-Cemig must provide services for the discovery, visualization, and recovery of geospatial data, which must be compatible with the OGC standards. The use of services based on the OGC standards allows SDI-Cemig to interact with other SDIs at different levels, such as the INDE (*Infraestrutura Nacional de Dados Espaciais* – National Spatial Data Infrastructure), the INSPIRE (Infrastructure for Spatial Information in the European Community), and the CGDI (Canadian Geospatial Data Infrastructure). For a new service to be considered compatible with the OGC standard, its operations must follow the specifications proposed in the documents provided by the OGC.

Although Figure 3 shows that the component *Product* is self-related, since a service may generate new data, SDI-Cemig has no processing service able to produce new geospatial data at first.

The SDI products will be described by *Metadata*, which are specified according to the *Metadata Geospatial do Brasil* (Geospatial Metadata of Brazil - MGB) profile (CONCAR, 2009). The MGB profile defines the elements existing in the metadata that describe the geospatial data to be introduced into the INDE.

The metadata may be used by the *Processing Tools* to help discover new geospatial data and services and to obtain relevant information on them, e.g., which features are offered by the services and in which format the geospatial data is being made available. In SDI-Cemig, the *Processing Tools* are the legacy systems and desktop applications that use the SDI's geospatial data and services. Cemig has several applications and legacy systems to process geospatial data that are very important in the company's processes.

The component *Connectivity* specifies how the *Processing Tools* interact with the SDI, which is possible by using a *Technology*. Cemig's legacy systems and desktop applications interact with SDI-Cemig by exchanging files in the XML format using the GML standard as schema. Besides using files, the desktop applications can interact with SDI-Cemig through web services in case they are supported.

SDI-Cemig specifies at least one policy for each type present in Table 1, except for *Governance* and *Business Agreements*, which have no policy defined yet. The policies will not be presented due to space constraints. However, some policies will be pointed out along the text.

3.1.2 Communities and Roles in SDI-Cemig

Besides the components in SDI-Cemig, the viewpoint Enterprise specifies the communities that make up the SDI and the possible roles they may play to reach their goals.

A community is a concept of RM-ODP and is a set of one or more entities that have similar behavior and seek to reach a given common goal (Linington et al., 2011). The behavior the communities may take on are described through roles to facilitate them being reused. In the case of SDI-Cemig, the possible roles the communities may take on were described by Hjelmager et al. (2008), Cooper et al. (2011), and Béjar et al. (2012), were adapted and unified by Oliveira and Lisboa-Filho (2015), and are used to specify the communities.

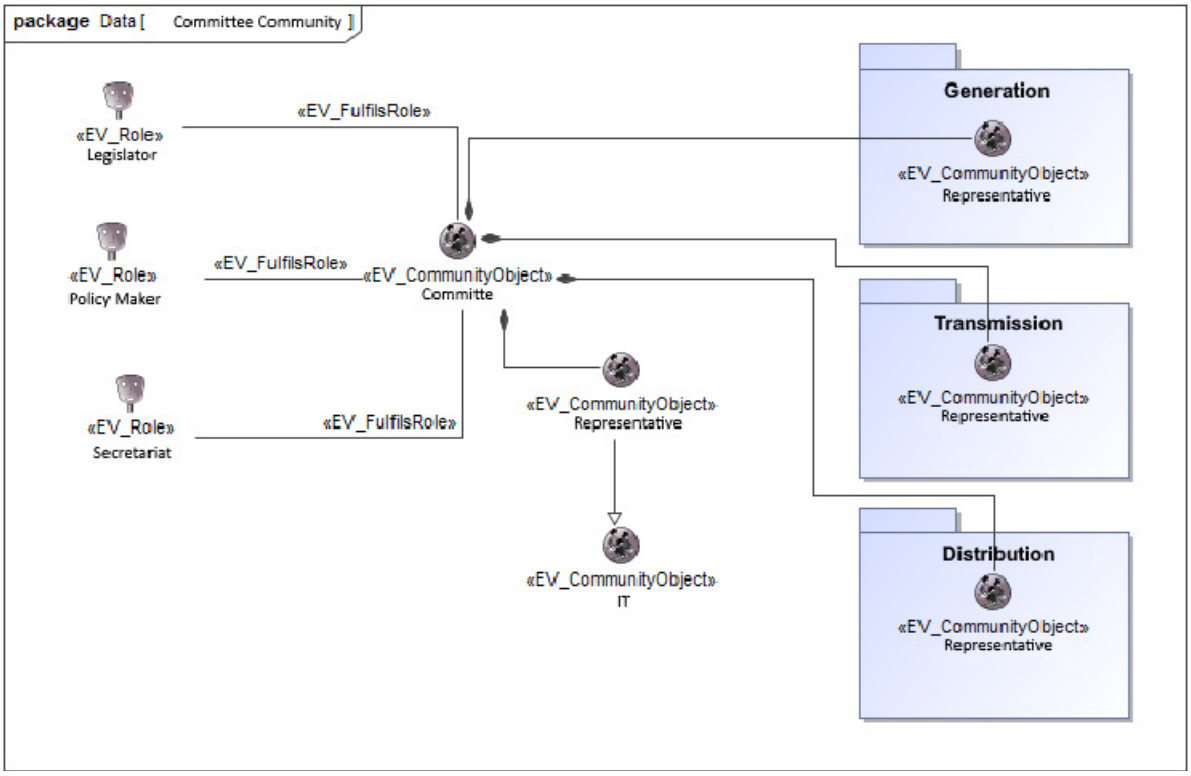


Figure 6: Community Committee and its respective roles.

According to Linington et al. (2011), a community is specified by the roles it can take on, its possible behaviors, the enterprise objects it uses, and the goal it must reach. This sub-section, however, details only the roles they may take on and whether these roles match the roles unified by Oliveira and Lisboa-Filho (2015).

Figures 6, 7, 8, and 9 present the communities identified in Cemig's environment and the roles they may take on when interacting with SDI-Cemig. In Figure 6, the community *Committee* is formed by members of different sectors at Cemig, represented by the communities *Representative*, such as *Information Technology (IT)* and the sectors *Generation*, *Transmission*, and *Distribution*, and its attribution is to define the working of certain processes carried out by these sectors. Hence, the *Committee* takes on the roles of *Legislator*, *Secretariat*, and *Policy Maker* and is responsible for all of SDI-Cemig's administrative area.

The community *GIS Analyst* (Figure 7) represents the IT individuals with positions homonymous to the community, who are responsible for carrying out and analyzing the procedures performed in a Geographic Information System (GIS) to manipulate geospatial data.

As shown in Figure 7, the *Geoprocessing Analyst* may take on the roles of *Data/Service Distributor*, *Data and Metadata Aggregator/Integrator*, and *Négociant*. The community is responsible for providing the geospatial data and services produced by the *Producers* in SDI-Cemig.

The community is also responsible for purchasing the geospatial data the users require, then acting as a *Négociant*. Finally, the *Geoprocessing Analyst*, when carrying out procedures on the geospatial data in a GIS, is able to generate new geospatial data or to expand existing data, thus taking on the role of *Data and Metadata Aggregator/Integrator*. Moreover, the *IT* will be in charge of creating and maintaining the catalogs of data and services made available by SDI-Cemig by using user-produced metadata.

Cemig has several sectors that act in the processes of electric energy generation, transmission, and distribution. The generation process consists in the generation of electricity through power plants and Cemig has hydroelectric, thermal, wind, and solar plants. Transmission consists in a network that carries the energy produced by the power plants to the large consuming centers. Finally, distribution is the network that serves energy to small- and medium-sized companies and to residential consumers (Leão, 2009).

The generation, transmission, and distribution groups are represented in Figure 8 by packages comprising all the sectors related to each group. Since there is a large number of sectors related to each group, they are represented by the communities *Generation*, *Transmission*, and *Distribution*. Besides these communities, each group has a G*eospatial Data Manager* and a *Representative*.

Each group has its *Spatial Data Manager* community, which is responsible for guaranteeing data consistency for each group, hence it takes on the role of *Database Administrator*. However, it must be pointed out that Cemig has a position called Database Administrator, although its role is different from the one defined by Cooper et al. (2011). At Cemig, the position Database Administrator is in charge of guaranteeing that the database and the hardware supporting it are in order.

The community *Representative* is a generic community used to illustrate the individuals that represent the interests of each group in the community *Committee*. Finally, each group has a homonymous community (*Generation*, *Transmission*, and *Distribution*) that represents the different sectors at Cemig that work directly or indirectly with the data of that group. The communities *Generation*, *Transmission*, and *Distribution* are considered *Official Production Agencies* since they are the main data producers in SDI-Cemig and since their sectors belong to Cemig. These communities are also responsible for publicizing the data they produce in the SDI, thus taking on the role of *A Producer that is its own Data/Service Provider*.

SDI-Cemig interacts with other communities besides those within Cemig itself by interacting with other SDIs and organizations, as shown in Figure 9. The community of the *Instituto Brasileiro de Geografia e Estatística* (Brazilian Institute of Geography and Statistics - IBGE) is the federal public organ that produces nationwide geospatial data, besides being responsible for defining the standards to be used by the other geospatial-data-producing organizations, thus taking on the role of *Producer*. The data produced by the *IBGE* are publicized through the INDE. SDI-Cemig interacts with the *INDE* and recovers the data available through web services, which makes the *INDE* a *Provider* of SDI-Cemig.

Besides the *INDE*, SDI-Cemig will obtain and publicize information to the *Sistema de Informações Geográficas do Setor Elétrico* (Geographic Information System of the Power Sector - SIGEL) belonging to the *Agência Nacional de Energia Elétrica* (National Electric Energy Agency - ANEEL). *ANEEL* is responsible for regulating and overseeing the Brazilian electric energy market to guarantee that the companies working in the country follow the regulations in effect. The *SIGEL* is a system that allows the visualization and obtention of some geospatial data made available by the utility companies to *ANEEL*. Therefore, *ANEEL* takes on the role of *User* in SDI-Cemig by recovering the

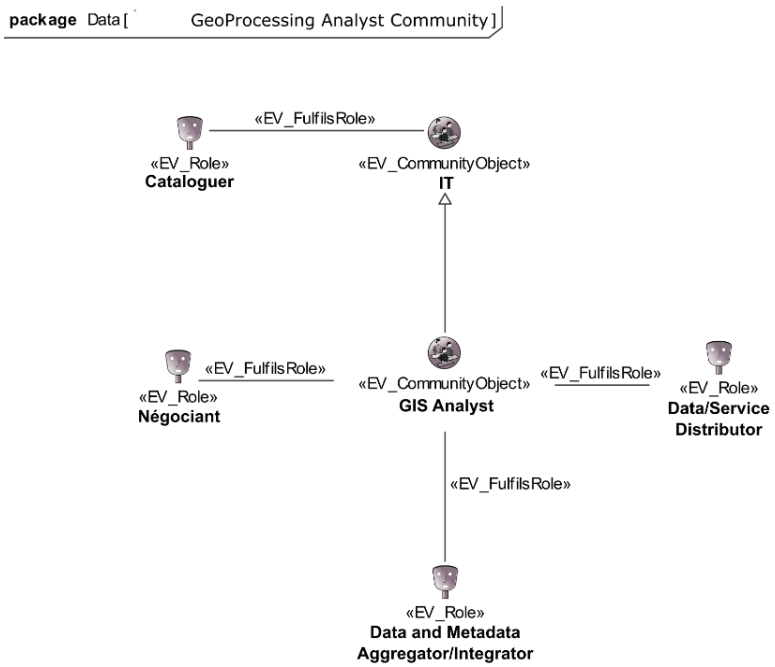


Figure 7: Geoprocesing Analyst Community and its respective roles.

data through the GeoPortal or through web services, while the *SIGEL* takes on the role of *Data Provider* by making available to SDI-Cemig the data provided to *ANEEL* by the other utility companies.

3.2 Viewpoint Information

As well as in the viewpoint Enterprise, the components defined by Hjelmager et al. (2008) for the viewpoint Information, shown in Figure 5 in sub-section 2.2, are identified in SDI-Cemig.

According to Linington et al. (2011), the viewpoint Information is responsible for "modeling the shared information that is handled by the system." Therefore, the invariant scheme of the geospatial database used in SDI-Cemig is modeled. The dynamic and static schemes are not modeled because SDI-Cemig, having only geospatial data, contains little or no dynamically generated data due to an action. When geospatial data are represented in alphanumeric format, comparing them to the original data to check whether the representation is consistent becomes difficult.

According to Hjelmager et al. (2008), the model presented in Figure 5 begins with the component Policies, which defines the basic geospatial data (layers) the SDI must have, besides allowing the link with the viewpoint Enterprise. The basic data SDI-Cemig has are described in the policies *Foundation*. It must be pointed out that much of the data in SDI-Cemig are related to the electricity generation, transmission, and distribution.

The members of SDI-Cemig may request new products (data and services) by opening a ticket with Cemig's helpdesk, being limited by the policies. Such tickets are considered the products' specifications (component *Product Specification*).

The Products are described by *Metadata*, which allows the users to assess whether the product meets their needs, besides facilitating searching for them. According to the policy *Legal Constraints* "*Adoção do Decreto de Lei Nº 6.666 – Uso do perfil MGB para a documentação de metadados geoespaciais produzidos em território nacional,*" the products in SDI-Cemig will be described using metadata documented following the specification of the MGB profile (CONCAR, 2009).

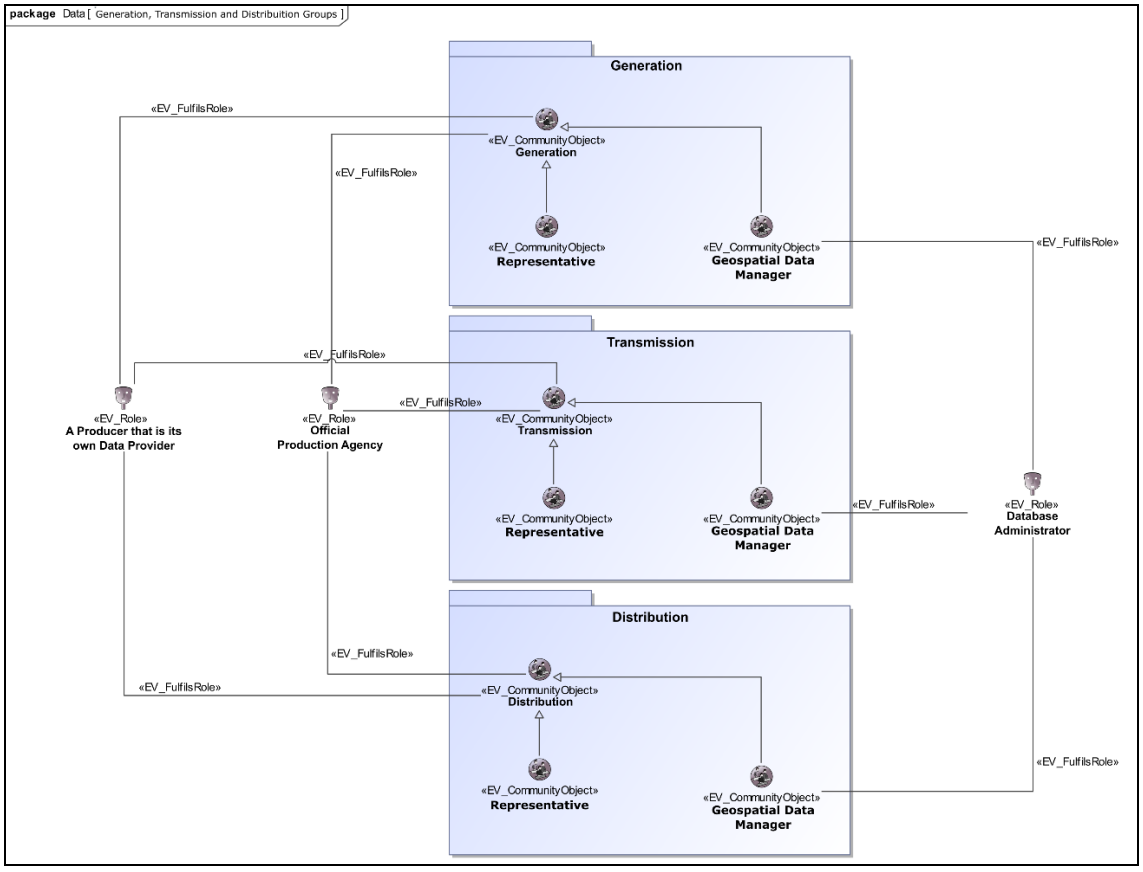


Figure 8: Groups Generation, Transmission, and Distribution with their respective communities and roles.

Both *Metadata* and *Products* will be recorded in a *Catalog* to aid in their discovery. The catalogs will be created according to the topics of the geospatial data offered by SDI-Cemig such as hydrography, generation, transmission, distribution, infrastructure, etc. According to the model in Figure 5, the data generate information based on pre-established knowledge. In SDI-Cemig, the data are used to generate information used by the different sectors at Cemig through reports and maps. Such information is generated based on the knowledge of employees specialized in geoprocessing, usually Geoprocessing Analysts.

3.2.1 Conceptual Database Modeling

According to Béjar et al. (2012), the policies of the type Foundation define the basic data and services the SDI must have. However, only the database description is not able to show the relationship among the data or how they will behave in the system, which is one of the goals the viewpoint Information aims to represent.

Figure 10 presents the conceptual scheme of the database adopted by SDI-Cemig. Due to space constraints, only the layers related to electricity generation, transmission, and distribution will be represented.

The UML class diagram extended with geographical and topological builders of the OMT-G (Borges, Davis Jr. and Laender, 2001) was used to create the scheme.

The package Distribution Grid has layers related to Cemig's regional distribution grid and layers that help manage this grid. The layer *Malha_Regional_Distribuicao* represents the limit of the distribution areas, which contain a headquarters (*Malha_Regional_Sede*) inside them. The business units (*Unidades_Negocio*) are areas defined according to the type of business Cemig intends to establish in a given region, which aids in planning and in the decision-making process. As well as the regional grid, the business units have a headquarters (*Unidades-Negocio_Sede*).

The area where Cemig can work in the state of Minas Gerais, negotiated with the state's government, is represented by the class *Areas_Concessao_Distribuicao*, while the class *Local_Cemig_Concessao* represents the area where Cemig is currently working. To help in the decision-making process, Cemig has divided the state of Minas Gerais into several regions called transmission regions (*Regionais_Transmissao*). As well as the distribution grid, the transmission regions are divided according to criteria that meet the company's business rules.

The packages Generation, Transmission, and Distribution contain the classes that represent the elements that make up the electric grid administered by Cemig. Cemig's electric grid nodes comprise structures, namely *Estruturas_LT* for *Generation*, *Estrutura_LT_230-500* for *Transmission*, and *Estrutura_LT_34-161* for *Distribution*. The classes

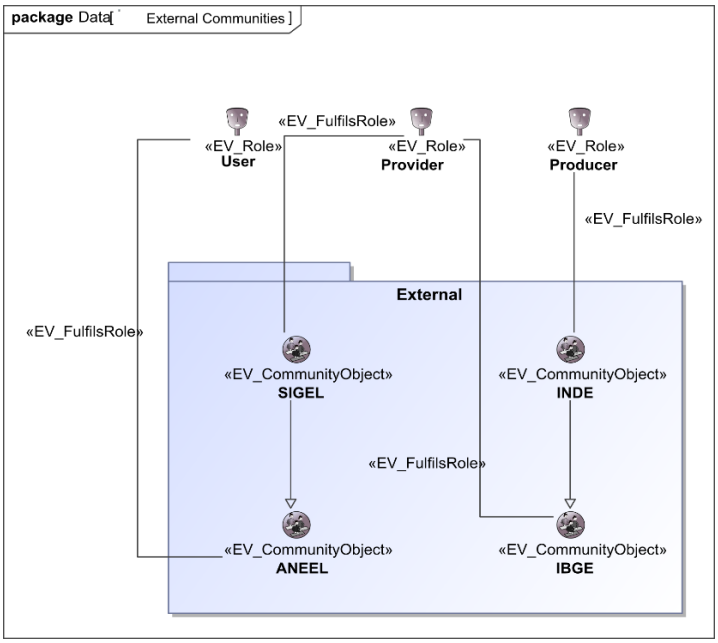


Figure 9: External communities that interact with SDI-Cemig.

Vao_LT, *Vao_LT_230-500*, and *Vao_LT_34-161* represent, respectively, the arcs of *Generation*, *Transmission*, and *Distribution.*

The structures that make up the *Generation* nodes comprise power plants, which can be hydroelectric, wind, or solar (*Usinas_Hidreletricas*, *Usinas_Eolicas*, *Usinas_Solares*, respectively), and by *Centrais_Geradoras_Hidreletricas*, *Subestacoes_Geracao*, and *Pequenas_Centrais_Hidreletricas*. Although it is said in the subsection 3.1.2 that Cemig owns thermal power plants, they are not considered, at the first moment, in the conceptual model.

In Transmission, the only structures that make up the network are the transmission sub-stations (*Subestacoes_Transmissao*). In *Distribution*, the structures comprise *Postes* (poles) and *Subestacoes_Distribuição*. The poles may have a transformer. *Generation*, *Transmission*, and *Distribution* have, respectively, the classes *Linhas_Transmissao*, *Linhas_Transmissao_230-500*, and *Linhas_Transmissao_34-161*. These classes are used to identify a portion of the network, which must comprise at least an arc and its respectively beginning and end nodes.

4 DISCUSSION OF RESULTS

The adapted ICA model proved appropriate to describe the viewpoints Enterprise and Information of SDI-Cemig. The differences found between the model and the specification are due to the specific characteristics of SDI-Cemig.

One such difference is that there are no geoprocessing services. In the viewpoint Enterprise, the lack of geoprocessing services impacts the component Product, which cannot be self-related.

In addition, the existence of the component Technology in ICA's formal model contradicts the goal of the viewpoint Enterprise in the RM-ODP framework, which is to describe the system's scope, policies, and requirements. This contradiction can be extended to the component *Connectivity*, however, further studies are needed to state that.

Also regarding the viewpoint Enterprise, during the specification of the actors in SDI-Cemig, the concentration of positions in the IT community becomes visible, which are responsible for providing data to SDI-Cemig, performing maintenance in smaller systems, negotiating new geospatial data, and creating new policies. Many of these responsibilities are beyond the scope IT should take on in SDI-Cemig.

Regarding the policies, the ones related to the type *Governance* have not been defined yet. Moreover, other types of policies have a small number of policies specified (usually a single policy has been specified for each type).

The viewpoint Information of SDI-Cemig has all the components specified by the adapted ICA model, with no need to change their behavior or semantics.

Although the adapted ICA formal model describes SDI at all levels and, thus, guarantees the basic concepts in the literature are contemplated in the specification phase, there is no description of how the model should be used. For instance, how many details are required to describe the components of the viewpoint Enterprise, or what could be considered a product specification?

5 FINAL CONSIDERATIONS

Using the adapted ICA formal model allows the key components of an SDI to be contemplated in the design phase, besides allowing a better understanding of the basic concepts such as the SDI structure, who the users will be and what roles they will take on when using an SDI, how the policies will impact the SDI development, etc.

The viewpoints Enterprise and Information in ICA's formal model properly describe these viewpoints in SDI-Cemig and, although the specification of a single corporate SDI does not ensure the model will be applicable at any corporate level, it does indicate the viewpoints Enterprise and Information in ICA's formal model can be applied to other corporate SDIs. Moreover, the present study may help other designers wanting to use ICA's model to specify new SDIs regardless of their level.

As future works, we intend to specify the viewpoint Computation in SDI-Cemig to verify whether it is in accordance with the viewpoint Computation specified in the adapted ICA model.

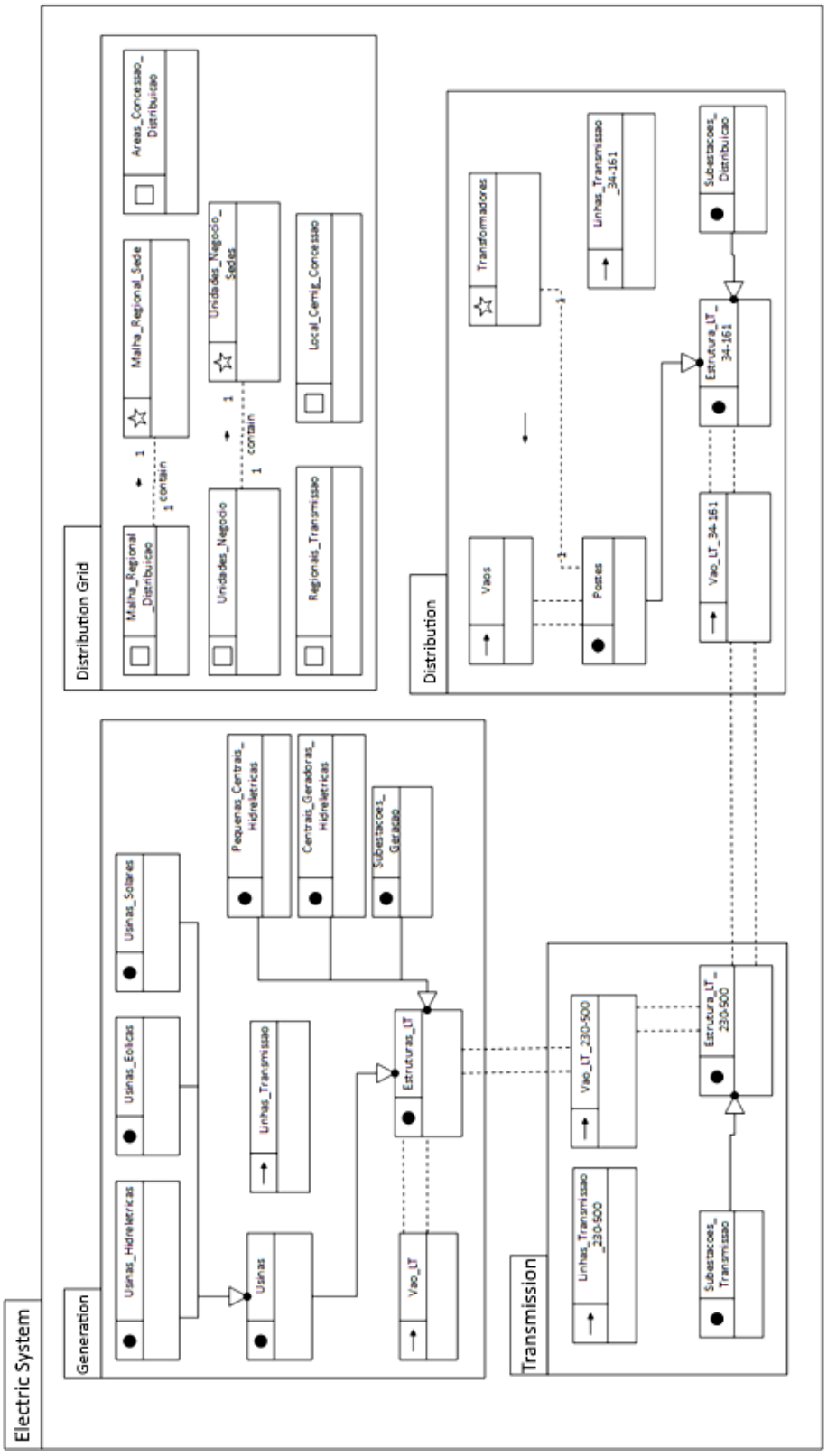


Figure 10: Layers related to the electric system and the distribution grid of the state of Minas Gerais from the conceptual scheme of SDI-Cemig's database.

ACKNOWLEDGEMENTS

This project was partially funded by the Brazilian research promotion agencies Fapemig and CAPES, along with Cemig Enterprise.

REFERENCES

Béjar, R., Latre, M. Á., Nogueras-Isso, J., Muro-Medrano, P., R., Zarazaga-Soria, F., J., 2012. An RM-ODP Enterprise View for Spatial Data Infrastructure. *Computer Standards & Interfaces*, v. 34, n. 2, p. 263-272.

Borges, K. A. V., Davis Jr., C. A., Laender, A. H. F., 2001. OMT-G: An Object-Oriented Data Model for Geographic Applications. Geoinformatica, v. 5, n. 3, p. 221-260.

CONCAR – Comissão Nacional de Cartografia, 2009. Perfil de Metadados Geoespaciais do Brasil (Perfil MGB). Available at: http://www.concar.ibge.gov.br/arquivo/perfil_mgb_final_v1_homologado.pdf.

Cooper, A. K., Moellering, H., Hjelmager, J., et al., 2013. A Spatial Data Infrastructure Model from the Computational Viewpoint. *International Journal of Geographical Information Science*, v. 27, n. 6, p. 1133-1151.

Cooper, A. K., Rapant, P., Hjelmager, J., et al., 2011. Extending the Formal Model of a Spatial Data Infrastructure to Include Volunteered Geographical Information. *25th Cartographic Conference (ICC)*.

Crompvoet, J., 2011. Spatial Data Infrastructure and Public Sector. Available at: http://www.spatialist.be/eng/act/pdf/20111107_sdi_intro.pdf.

Farooqui, K., Logrippo, L., De Meer, J., 1995. The ISO Reference Model for Open Distributed Processing: and introduction. *Computer Networks and ISDN Systems*, v. 27, n. 8, p. 1215-1229.

Harvey, F., Iwaniak, A., Coetzee, S., Cooper, A., K., 2012. SDI past, present and future: a review and status assessment. *Spatially Enabling Government, Industry and Citizens*.

Hjelmager, J., Moellering, H., Cooper, A. K., et al., 2008. An Initial Formal model for Spatial Data Infrastructure. *International Journal of Geographic Information Science*, v. 22, n. 11-12, p. 1295-1309.

Leão, R., 2009. GTD – Geração, Transmissão e Distribuição de Energia Elétrica. Universidade Federal do Ceará, Centro de Tecnologia, Departamento de Engenharia Elétrica. Available at: http://www.clubedaeletronica.com.br/Eletricidade/PDF/Livro%20GTD.pdf. (in Portuguese).

Linington, P. F., Milosevic, Z., Tanaka, A., Vallecilo, A., 2011. *Building Enterprise Systems with ODP: An Introduction to Open Distributed Processing*. CRC Press.

Nebert, D., D., Techinical Working Group Chair GSDI, 2004. Developing Spatial Data Infrastructures: The SDI Cookbook. V.2. *GSDI – Global Spatial Data Infrastructure*. Available at: http://www.gsdi.org/docs2004/Cookbook/cookbookV2.0.pdf.

Oliveira, I. L., Lisboa-Filho, J., 2015. A Spatial Data Infrastructure Review – Sorting the Actors and Policies from Enterprise Viewpoint. *Proceedings of the 17th International Conference on Enterprise Information Systems*, v. 17, p. 287-294.

Rajabifard, A., Williamson, I., P., 2001. Spatial Data Infrastructures: concept, SDI hierarchy and future directions. *Proc. of GEOMATICS Conference*, p. 10.

Raymond, K., 1995. Reference Model for Open Distributed Processing (RM-ODP): Introduction. *Open Distributed Processing*, p. 3-14.

Modeling Variability in Software Process with EPF Composer and SMartySPEM: An Empirical Qualitative Study

Jaime W. Dias and Edson OliveiraJr
Informatics Department, State University of Maringá, Maringá-PR, Brazil
jaimewdias@gmail.com, edson@din.uem.br

Keywords: Annotative Approach, Compositional Approach, Eclipse Process Framework, Empirical Qualitative Study, SMartySPEM, Variability.

Abstract: Nowadays, organizations are increasingly seeking to customize their software processes according to the market needs and projects experiences. Therefore, a systematic way to achieve such an objective is the Software Process Line (SPrL) technique, in which each member is a customized software process derived from a set of similarities and variabilities of process elements. Compositional and annotative approaches are most referenced for variability management. In this sense, the objective of this paper is to present a comparison between compositional and annotative approaches targeting variability representation capabilities from the point of view of practitioners and academic experts. The Eclipse Process Framework by means of the EPF Composer tool represents compositional approach, whereas SMartySPEM represents the annotative approach. The obtained results provided initial evidence that the annotative approach (SMartySPEM) takes advantage over the compositional approach (EPF).

1 INTRODUCTION

Software development companies need to quickly develop software with quality, which makes them adopt software reuse techniques (Aleixo et al., 2013). Such a reuse aims at improving productivity, reducing coding effort and increasing quality as the software has been already verified and validated.

The current scenario of competitiveness has led software companies to seek solutions beyond software reuse in order to reduce time-to-market and increase return on investment (ROI). Thus, software development projects need to be tailored according to the needs of a company and its development domain (Garcia-Borgonon et al., 2014). Therefore, the Software Process Line (SPrL) technique (Rombach, 2005) is an alternative for process customization based on similar and variable process elements. Such elements can be defined based on the concept of variability, from the Software Product Line (SPL) technique (Linden et al., 2007).

Currently, there are several tools and languages for software process modeling (Garcia-Borgonon et al., 2014), as well as approaches that guide such modeling, as for instance, compositional, annotative, transformational and model-driven (Kästner, 2010; Kästner et al., 2008; Kästner and Apel, 2008). Each approach is used in specific ways to represent variability among process elements (Aleixo et al., 2013). Therefore, this work presents a qualitative empirical study comparing the compositional approach, represented by the Eclipse Process Framework (EPF) and its Composer tool, and the annotative approach, represented by the SMartySPEM (OliveiraJr et al., 2013) approach for variability representation in software process elements.

2 BACKGROUND

This section presents essential concepts of software process lines, variability and the OpenUP-based SPrL, used in our qualitative study.

2.1 Software Processes and Process Lines

A software process can be defined as a set of techniques and technologies to support, evaluate and improve software development activities. The need to specifying software processes arises from the fact that the products quality can be directly influenced by the process adopted for their productions (Chemuturi and

Dias, J. and OliveiraJr, E.
Modeling Variability in Software Process with EPF Composer and SMartySPEM: An Empirical Qualitative Study.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 283-293
ISBN: 978-989-758-187-8

Cagley, 2010). The ISO/IEC 15504 standard (ISO, 2012) defines a process as a set of activities that are interrelated or interacting to transforming inputs into outputs. This set of activities serves as a guide for those who will be responsible for the process execution and monitoring.

A software development process has four basic steps (Sommerville, 2015): specification of the software features and premises for its development; design for constructing the software according to its specifications; validation to ensure that the software meets the users needs; and evolution in order to accommodate prospective necessary modifications.

Modeling of such steps are essential for a complete understanding of the process (Garcia-Borgonon et al., 2014). Basic elements and concepts are essential for software process modeling: Role, which describes how people act in the process and their responsibilities; Task, which is an action performed by a role for executing or monitoring a project; Activity, which is a set of tasks that lead to produce/consume one or more controlled quality artifacts; Artifact, which represents the result of a task; and Process, which is an organized collection of activities.

Basic elements of a software process are essential to enable process tailoring and customization, which is currently an important research topic from the academic and industrial point of view (Martínez-Ruiz et al., 2012; Kalus and Kuhrmann, 2013; Carvalho et al., 2014). Therefore, the term Software Process Line (SPrL), proposed by Rombach (Rombach, 2005), has been considered in the last years, suggesting the adoption of important concepts from software product lines, such as similarities and variabilities. An SPrL provides techniques and mechanisms for modeling existing similarities and variabilities in a family of software processes, as well as the derivation of customized software processes that meet the specific needs of a given software development project (Rombach, 2005; Aleixo et al., 2011).

SPrLs may contain variation points, which are process elements that can be instantiated in different ways. For each variation point there are variant elements, which can be selected to resolve a specific variation point (OliveiraJr et al., 2013). Figure 1 illustrates an excerpt of an SPrL, in which the similar (mandatory) part is composed of the role `Developer` and two tasks: `Design the solution` and `Implement solution`. Such a figure also illustrates two variabilities: (i) the inclusion of developer tests practices (`Implement developer tests` and `Run developer tests`), and (ii) the inclusion of integration and creating of a build (`Integrate and create build`). This excerpt can generate four distinct process instances: (i) with common elements only by discarding the variabilities, keeping mandatory process elements, (ii) only resolving the variability concerned with `Developer test`, (iii) only resolving the variability concerned with `Integration continues`, and (iv) resolving both variabilities considering all the elements.

2.2 The OpenUP-based SPrL

The OpenUP process complies with the principles of the Agile Software Development Manifest, thus it can be taken as an agile version of the Unified Process (UP), meeting UP good practices. OpenUP is an iterative and incremental approach, with no specific tools.

Several activities of the OpenUP are optional, however, it does not define which elements of the processes vary from the SPrL point of view. Thus, in the work of (Aleixo et al., 2011), Aleixo et al. presents excerpts of the OpenUP modeled as an SPrL, defining similar features and variabilities. To do so, three real research and development projects based on OpenUP in addition to experienced practitioners were involved in the definition of such an SPrL. The first project dealt with the development of a software system to audit telephone networks, the second project involved the development of a module of a distributed system and the third project involved the implementation of an integrated academic and administrative management. As a result of this analysis it was identified 586 features, from which: 273 mandatory features, 239 optional features and 74 alternative features.

3 VARIABILITY MODELING IN SPRL WITH COMPOSITIONAL AND ANNOTATIVE APPROACHES

The success of an SPrL depends on the accuracy of its variability management activity (Rombach, 2005). Thus, such a management is a key requirement in the development of SPrLs to provide support to specification, implementation, variability resolution and customized processes generation. Variability management defines how common and variable artifacts are represented and treated in order to generate process instances from an SPrL.

There are different approaches and techniques for variability management in the literature. They can be classified as (Galster et al., 2013): compositional, annotative, transformational, and model-driven. This study concerns on compositional and annotative ap-

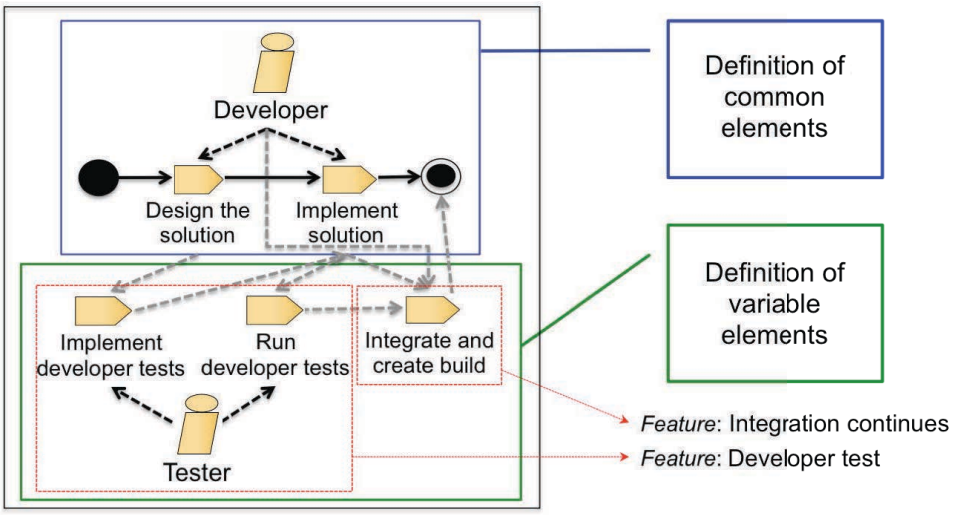


Figure 1: Excerpt of an SPrL (Aleixo et al., 2013).

proaches as they are highly referenced in the literature.

3.1 Compositional Approach

The compositional approach supports modularity of physical features, thus generation of products occurs by means of selecting and composing modules that implement features of desired products (Galster et al., 2013). Therefore, development techniques are highlighted: Feature-Oriented Programming (FOP) (Lee and Kang, 2013) and Aspect-Oriented Programming (AOP) (Kiczales et al., 1997). FOP provides support to the similarities and variabilities be modularized and each feature implemented in a distinct module. Such a module is an increment in the functionality of a base system (step-wise refinement).

An example of software process modeling using the compositional approach is the Eclipse Process Framework (EPF), which allows editing, configuring and publishing of software processes. EPF persists process information according to the Unified Method Architecture (UMA) meta-model, developed based on the SPEM 1.0. Subsequently, UMA inspired the creation of the SPEM 2.0.

Figure 2 presents the EPF framework main parts using the EPF Composer, a process modeling tool based on the EPF framework, as follows:

- Method Content: standardizes representation and manages reusable component libraries. It defines roles, tasks, work products and their relationships;
- Process: determines the sequence of phases, iterations and activities, and defines when tasks are performed;

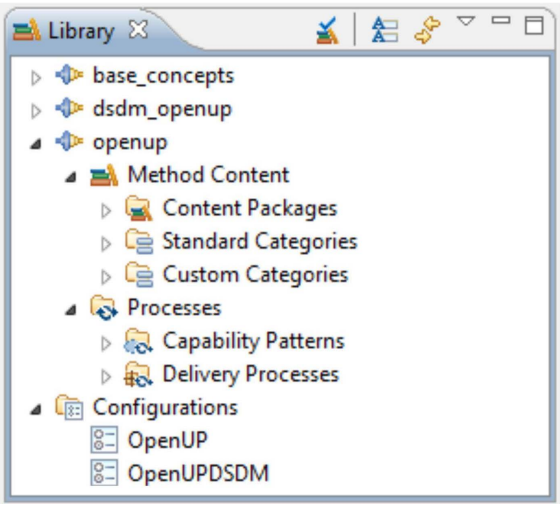


Figure 2: EPF Excerpt Illustrating Method Content, Process, Plug-ins and Configurations using the EPF Composer Tool.

- Plug-ins: represent a set of Method Content and process packages, allowing process customization;
- Configurations: selects a subset of Method Contents to form a specific process by publishing it in HTML or exporting it to MS Project or XML.

The EPF framework allows representing artifacts variability in order to control the evolution and reuse of software processes. There are four possible types of variability in EPF, which are:

- Contributes - contributing (variability) element adds to a base element;
- Replaces - replacing (variability) element replaces parts of the base element;

- Extends - extending (variability) element inherits characteristics of the base element. The base element is unchanged; and
- Extends-Replaces - combines the effects of extends and replaces variability, allowing one to selectively replace specific attributes and relationships of the base element. Extending-replacing (variability) element replaces values in the base element that have been redefined in the extending-replacing element.

3.2 Annotative Approach

The annotative approach provides the use of preprocessor directives to annotate code snippets associated with a particular feature (Kästner et al., 2008). The C and C++ languages already support preprocessor directives. Products generation occurs by defining the value of the symbolic constant of pre-processing directives associated with selected features, before prebuilding, in order to define the presence of the features snippets selected in the generated product. Just as in Java annotations, which provide the option of using metadata over code that can be later interpreted by a compiler or pre-compiler that performs predefined tasks. Another way of annotation is the UML stereotypes that add semantics to existing elements with no changes in their meta-model.

Stereotype-based Management of Variability for the SPEM meta-model (SMartySPEM) (OliveiraJr et al., 2013) is an approach that provides the separation of elements and their management by using a visual annotation that associates notes and stereotypes to each type of process elements variability.

The SMartySPEM approach aims at supporting the identification and representation of variability in processes elements modeled with SPEM. To do so, SMartySPEM introduces the profiling mechanism based on the SMarty approach (OliveiraJr et al., 2010) for representing variability in SPEM modeled elements with specific stereotypes. SMartySPEM is composed of a UML 2.0 compliant profile, the SMartySPEMProfile, with the following stereotypes (OliveiraJr et al., 2013):

- ≪**variability**≫ - represents the concept of variability (UML note);
- ≪**variationPoint**≫ - represents the concept of variation point (VP icon), in which a variable process element provides a set of choices for customizing a software process;
- ≪**mandatory**≫ - represents compulsory process elements (MDT icon), present in every customized software process;
- ≪**optional**≫ - represents an optional variant;
- ≪**alternative_OR**≫ - represents inclusive variants (OR icon) to resolve a variation point;
- ≪**alternative_XOR**≫ - represents mutually exclusive variants (XOR icon) to resolve a variation point;
- ≪**mutex**≫ - represents mutual exclusion constraint (dependency relationship) among variants; and
- ≪**requires**≫ - represents that a given variant requires the presence of another variant (dependency relationship).

Figure 3 presents an excerpt of an `Architectural Analysis` activity modeled according to SMartySPEM. It contains the variation point `Analysis Class` with three related inclusive variants: `Control Class`, `Entity Class` and `Boundary Class`. Another variation point is `Architectural Analysis` with three variants: `Identifying Common and Special Requirements`, `Identifying Obvious Entity Classes` and `Develop Business Type Model`. `Architect` is a mandatory element for performing the `Architectural Analysis` activity. There are optional variants, such as, `Use-Case Model` and `Architecture Description`, which may or may not be present in derived processes (OliveiraJr et al., 2013).

4 EPF COMPOSER VS. SMartySPEM: A QUALITATIVE STUDY

This study aims at comparing compositional and annotative approaches represented, respectively, by EPF Composer and SMartySPEM. The comparison criteria to be adopted were proposed in the works (Kästner, 2010; Kästner et al., 2008; Kästner and Apel, 2008), namely: modularity, traceability, error detection, granularity, adoption and systematic variability management. These criteria were also used in the qualitative study of Aleixo et al. (Aleixo et al., 2012).

The criterion of `modularity` aims at analyzing the modularization degree of processes elements associated with specific features, enabling a better understanding and facilitating the maintenance and evolution of SPrLs. `Traceability` allows one to analyze how difficult is viewing and mapping of all process elements along with their associated features. The `error detection` criterion analyzes how efficient is

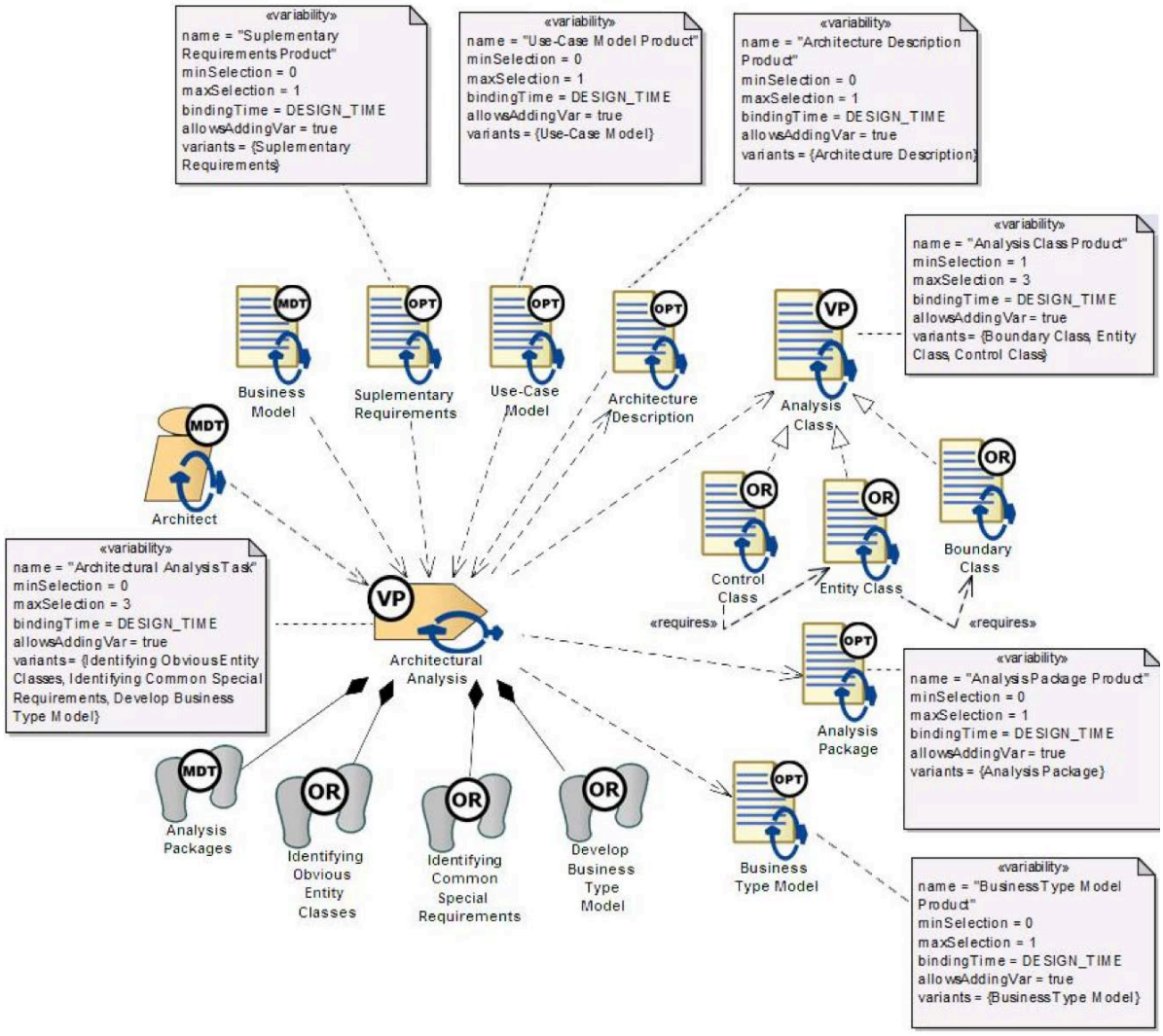


Figure 3: An Excerpt of Architectural Analysis Modeled with SMartySPEM (OliveiraJr et al., 2013).

an approach for identifying cohesion errors in the definition of an SPrL and its elements, as well as its derived processes. `Granularity` evaluates the approach support for representing variability in coarse and fine granularity, considering the division of the process in small or large parties. The `adoption` criterion discusses the difficulty of adopting an approach, analyzing necessary pre-knowledge to be taken to applying such an approach. `Systematic variability management` analyzes the mechanisms provided by an approach for specifying variability.

4.1 Objective

The aim of this study is to obtain experts' feedback with regard to the criteria for each approach. Then, compare such approaches in order to be able to draw initial evidence based on each criterion.

4.2 Planning

As study object we used the OpenUP-based SPrL from Section 2.2. As such an SPrL is too large with hundreds of features, we chose only the `Requirements Specification` feature.

Participants were given eight documents: a study consent stating the confidentiality of the responses; a characterization questionnaire to measure the participant's experience; a document with main concepts of SPrL; a document about the compositional approach using the EPF framework; a document about the annotative approach using SMartySPEM; a document with the modeling of the `Requirements Specification` feature in EPF (Figure 4); and a

document with the modeling of the `Requirements Specification` feature in SMartySPEM (Figure 5). After the trainning session, each participant was given two questionnaires containing six questions. Each question with regard to each criterion for each approach. Then, participants should answer the following questions, replacing TYPE_OF_APPROACH with "Compositional" or "Annotative":

1. The `modularity` criterion measures the quantity of modules (groups of process elements) necessary for representing an SPrL, thus is it possible to measure the modularity of the TYPE_OF_APPROACH approach?
2. The `traceability` criterion allows analyzing the visualization and mapping difficulty of all process elements along with their features, thus is it possible to visualize the traceability of the TYPE_OF_APPROACH approach?
3. The `error detection` criterion analyzes how efficient is an approach to identify cohesion errors in the definition of an SPrL and its elements, as well as the derived processes from the SPrL. Is it possible to detect cohesion errors in the TYPE_OF_APPROACH approach?
4. The `granularity` criterion aims at evaluating the approach support for representing variability in coarse and fine granularities (level of abstraction), thus considering the process division in small or large parts, is it possible to evaluate the granularity at the TYPE_OF_APPROACH approach?
5. The `adoption` criterion discusses the difficulty of adopting an approach, analyzing the amount of previous knowledge for applying an approach, thus have you experienced any difficulties for understanding the TYPE_OF_APPROACH approach?
6. The `systematic variability management` analyzes the provided mechanisms of an approach for specifying variability. Do you consider sufficient the variability mechanisms of the TYPE_OF_APPROACH approach?

Participants of this study were carefully selected based on their experience with software process. In average, each participant had ten years of experience working with software processes. Thus, twelve participants were invited for this study, from which one participant was invited for a pilot study for evaluating our instrumentation, thus his/her results were discarded. Amongst the participants are researchers and practitioners from the State University of Maringá (UEM), Federal University of São Carlos (UFSCar), Federal Technological University of Paraná (UTFPR) and University of York, all of them with masters or Ph.D.

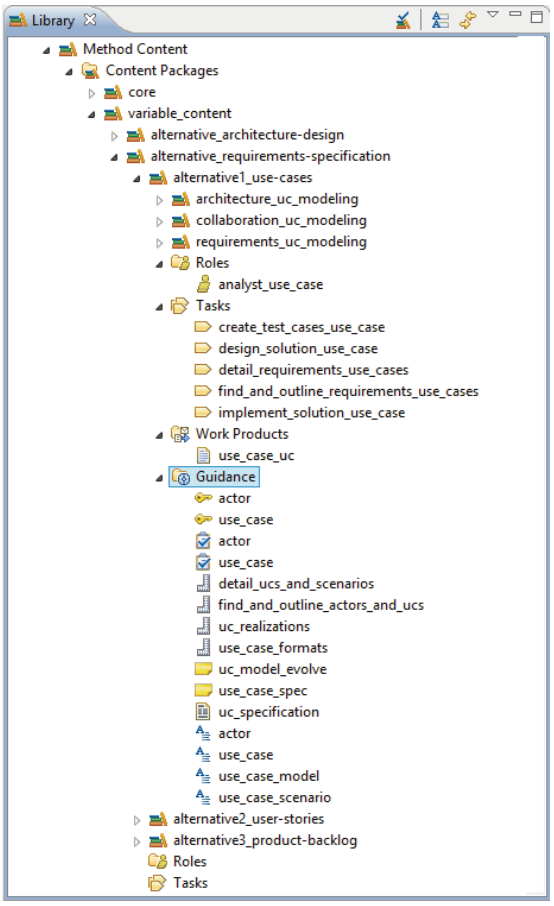


Figure 4: The OpenUP-based SPrL Requirements Specification Using the EPF Composer.

The realization of a pilot study led us to changes in the study planning as, for instance, in the study duration time. At first, we distributed all the materials and asked the pilot study participant to answer the two sets of six questions. This whole process took about two hours and forty minutes, making the participant tired and bored. Thus, we decided two divide it into two parts, each part in different days: the first one involving only one of the approaches and the second one the other approach. Therefore, each part of the study took no longer than 50 minutes.

Another necessary change based on the pilot project results was removing the uniformity criterion that aimed at assessing the technology or independent meta-model. As the two approaches depend on specific tools (EPF Composer and a UML Tool), we understand that such a criterion could be a potential threat to our study as the participants do not have enough experience with these tools, especially EPF Composer.

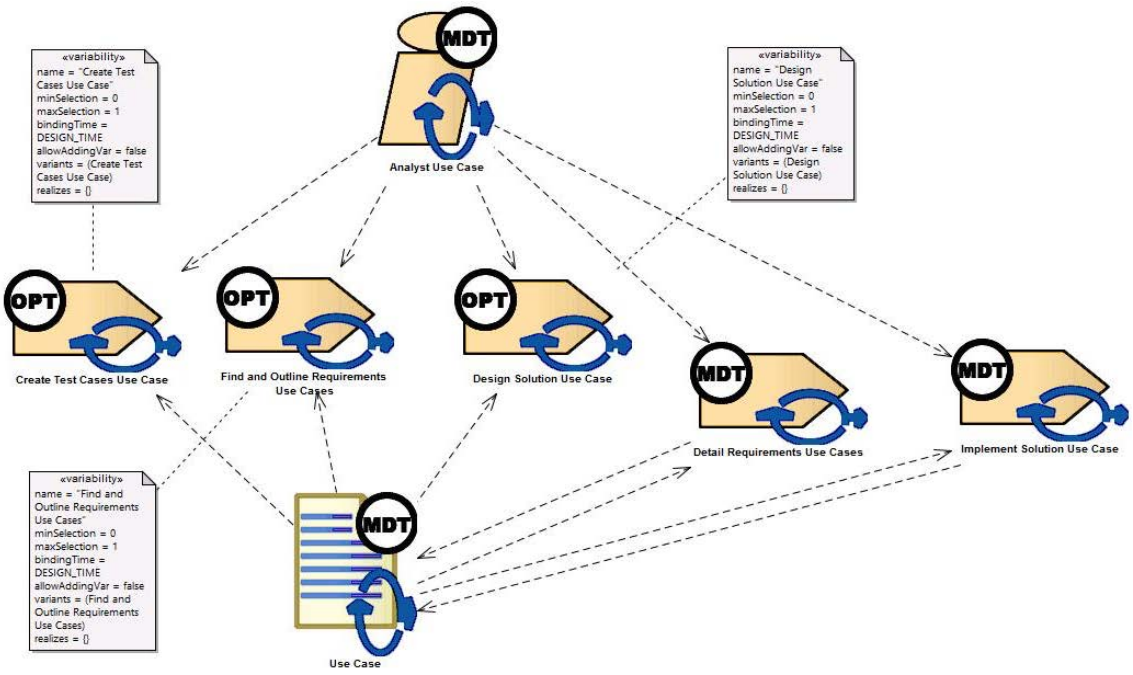


Figure 5: The OpenUP-based SPrL Requirements Specification According to SMartySPEM.

For reducing the threats to validity of the study, for each participant we changed the order of the evaluated approach. Then, five participants firstly received the annotative approach (SMartySPEM), whereas six participants firstly received the compositional approach (EPF Composer).

Responses from the experts were qualitatively analyzed using Grounded Theory (GT) procedures (Corbin and Strauss, 2008). The GT approach is based on coding concepts. Coding allows one to assign codes or labels for text snippets (Open Coding), which can be grouped and classified (Axial Coding) according to an idea expressed in order to elucidate a given phenomenon (Corbin and Strauss, 2008). As a result, such codings enabled the creation of a conceptual model. During the encoding process two categories were created "Feasible Criteria" and "Non-Feasible Criteria", thus grouping answers with respect to each criterion. An assistance tool for qualitative analysis was used, named Dedoose. Such a tool allows one to import a spreadsheet with results, then creating codings in specific excerpts of the answers. The created codings can relate to one another in a significant level. Then, after creating all codings one can produce graphical representations, such as, charts with different results visualizations.

4.3 Results

Figure 6 presents the results of this study as a comparison of the answers for each criterion of EPF Composer and SMatySPEM. The more centralized the line is (red or blue) in the chart the worse is the evaluation of a given criterion. As a result, the compositional approach was better evaluated for criteria `Modularity` and `Error Detection`, whereas the annotative approach was better evaluated for criteria `Traceability`, `Granularity`, `Adoption` and `Systematic Variability Management`.

Modularity in the annotative approach obtained nine positive evaluations, for example, expert #3 answered that "*...it is possible to measure such a modularity, as well as presenting the relationship between process elements in a module. Such a view contributes to the comprehension of what happens in each process module and, consequently, enables better understanding the tasks of a given derived process...*". However, it took two negative answers. Expert #5 stated that "*...SMartySPEM does not appear to enable an effective organization of the process elements of an SPrL. It is possible to establish the relations, but not distribute them so efficiently...*". Expert #4 answered that "*...as the process model grows, measuring modularity becomes more complex as it depends on the visualization (annotations) of all process elements in a diagram...*". In fact, for large SPrLs, visualization of the modules is jeopardized due to the amount of elements and annotations in a same diagram. On the other hand, the compositional approach took 100% of positive answers as it allows grouping the SPrL com-

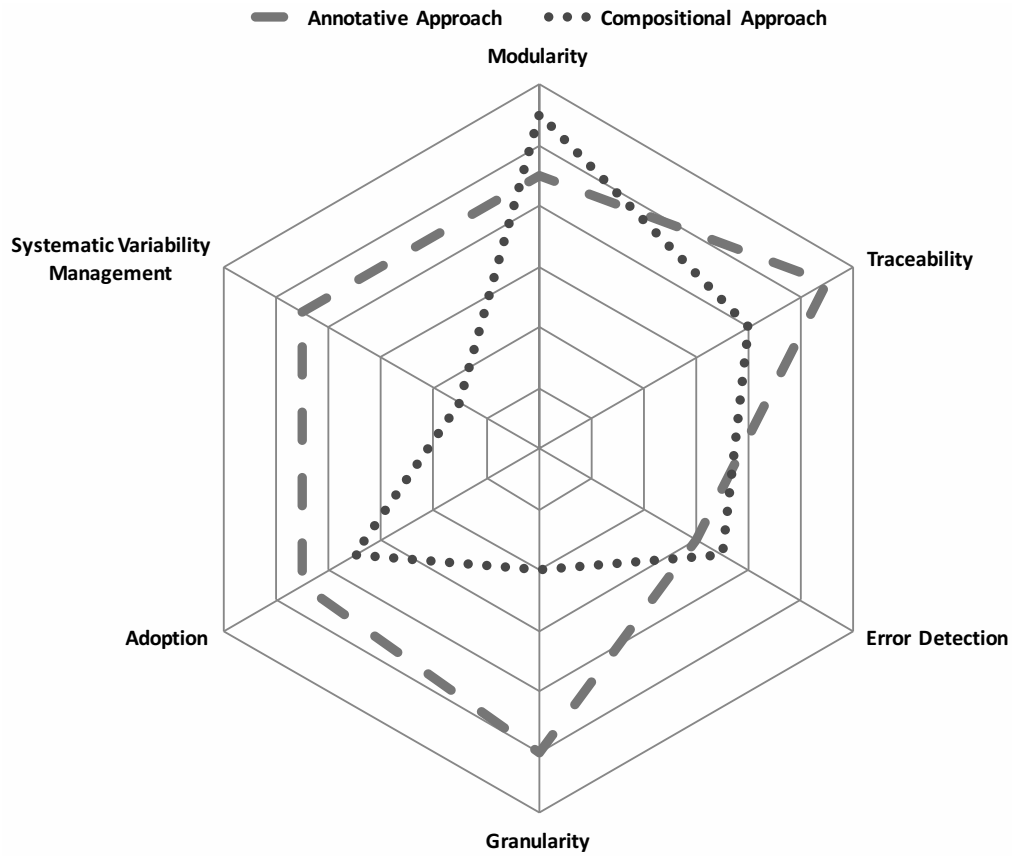


Figure 6: Spider Chart for Evaluating Compositional and Annotative Approaches based on the Defined Criteria.

ponents in hierarchical packages, with a better visualization of the modules as, for instance, reported by the expert #5 "*...EPF Composer is able to encapsulate the compositional elements in a satisfactory way. The use of plugins, method contents and configurations enables reusing such modules in an agile and intuitive way...*".

The **error detection** criterion was not satisfactory for both compositional and annotative approaches, as none of them provides verification activities. However, the compositional approach took a subtle advantage as the EPF Composer checks models at specific process publication (derivation) time as stated by expert #11 "*...it has some kind of support mechanisms for error detection. Some filters, as in the association with variabilities avoid wrong contributions/replacements/extensions of incompatible process elements...*". On the other hand, in the annotative approach, as it is an UML extension and it allows the relationship among several different elements, it is difficult to detect errors, as said by expert #8 "*...Because there is no automation of the approach, the only way to detect errors would be based on reviews and in-depth analysis comparing the process modeling with the description of the SPrLs elements...*".

Traceability obtained all the positive ratings for the annotative approach as it has a very visual appeal, facilitating the identification and traceability among process elements, as well as the `realizes` meta-attribute of the stereotypes ≪variability≫ and ≪variationPoint≫, which represents a collection of low-level models that realizes a given variability. An example is the statement from expert #8 "*...traceability is perceived by means of dependencies among process elements, as well as the* `realizes` *meta-attribute from variabilities or optional and variation point elements...*". For the compositional approach eight positive answers were given as such an approach allows relationships among process elements. However, various negative aspects were pointed out as the difficulty of graphically visualization of the relations between elements, and hiding of variability elements as stated by the expert #5 "*...The user accessing the published process, in html format, generated by the EPF Composer can not visualize inherited packages and configurations of a given element, when its variability allows such a visualization...*".

The granularity criterion had much more positive

evaluations for the annotative approach than the compositional due to the possibility of modeling different types of diagrams representing different abstraction levels of SPrLs. For instance, flow of activities with Activity elements for coarse-grained granularity and diagram description of a process with activities as fine-grained granularity. We can observe these examples from the expert #2 answer excerpt as follows: "*...the annotative approach clearly presents two levels of abstraction encompassing both fine and coarse-grained granularity, as the Activity process element provides such an abstraction. Abstraction levels allow and facilitate the visualization of the whole process, as well as make it easier identifying the variation of each of the process elements in lower-level abstraction levels...*". In the compositional approach this criterion obtained seven negative evaluations, because of the approach consider only one low abstraction level, not allowing a wide view of the SPrL. Examples of this characteristic are the expert #5 answer "*...the compositional approach has fine-grained granularity as the EPF Composer is aimed at configuring all aspects of the process by structuring it in detail...*" and the expert #1 answer "*...There is no higher abstraction level in order to compare variabilities of an SPrL...*".

The adoption criterion obtained nine positive evaluations for the annotative approach against seven for the compositional approach. We understand that this is straightforwardly related to the influence that the annotative approach because of the UML standard notation as we can see in the statements of expert #4 "*...the level of difficulty on applying the approach is low...*", expert #10 "*...I had no difficulty at understanding the approach as the number of elements to be learned for adopting the approach is not very large. I would adopt the approach without much effort...*" and expert #11 "*...I believe that the difficulty of adopting this approach is low, as its stereotypes, comments and elements are quite similar to UML representations...*". On the other hand, in the compositional approach, behavior complexity and use of variability types influences the adoption of such an approach, as well as difficulties in using the EPF Composer tool to manipulate the SPrL elements, as reported by the expert #2 "*...I had difficulties mainly in the elements that define variability. Also, such a tool needs extra effort on creating and maintaining process elements...*".

For **systematic variability management** the compositional approach was highly unsatisfactory, according to the experts evaluation, because of its variability mechanisms, because of the visualization of the applied variabilities to process elements. In addition, the experts judged that the four variability types are not enough. It can be noted in the expert #10 statement "*...as I have knowledge of other approaches for variability representation in software product lines, I understand that different variability mechanisms could be added to the EPF Composer. Thus, I judge such a set of mechanisms insufficient...*", and in the statement of the expert #9 "*...some new mechanisms of variability types are necessary as, once there is the extend mechanism, it must exist include, mandatory and optional mechanisms...*". In the annotative approach variability mechanisms had more positive evaluations as such an approach has specific stereotypes for representing variability of each process element type. Exemplary statements excerpts from experts are: expert #1 stated that "*...I believe the variability mechanisms are sufficient as stereotypes allow the specification of different variability types...*"; expert #3 said "*...I understand the SMartySPEM variability mechanisms are sufficient to significantly demonstrate the types of relationships between a variation point and variants...*"; and expert #10 said "*...I consider the variability mechanisms adequate for representing mandatory, alternative and optional elements. Furthermore, the mechanisms allow clearly visualize where variation points and variants take place in a non-ambiguous basis for those who use process models...*".

4.4 Validity Evaluation

Results validity evaluation is an essential issue of empirical studies (Wohlin et al., 2000). We discuss the main threats relevant to our study, as follows:

Internal Validity. Tasks performed by experts were conducted in a similar manner except the order of the application of the study objects and questionnaires, which were random. Experts were trained on the basics of SPrLS and variability in compositional and annotative approaches using EPF Composer and SMartySPEM. We reduced the fatigue effects allowing the experts to answer the questionnaires in at most fifteen days;

External Validity. Features related to Requirement Specification of the OpenUP-based SPrL were used during both the training sessions and the empirical study. This could jeopardize the external validity, thus we tried to use original and technical documents of the OpenUP;

Conclusion Validity. The major threat to conclusion is related to the sample size, eleven experts. However, prior knowledge of such experts is significant. Therefore, we understand that for a qualitative study in which grounded theory procedures, such as Coding, were established eleven experts is a satisfac-

tory number;

Contruct Validity. This study was planned based on a pilot project carried out for evaluating the instrumentation and its duration time for the application of questionnaire. Although the knowledge level required on software process and variability is essential, participants presented a high level of expertise.

5 RELATED WORK

Compositional and annotative integration and/or comparative studies have been carried out in the literature for several different domains, such as embedded systems and software product lines (Kästner and Apel, 2008; Ferreira Filho et al., 2013; Behringer, 2014). For software process lines or process tailoring/customization there is a lack of such a study type.

The study of Aleixo et al. (Aleixo et al., 012a) is the most direct related work to ours in the literature, in which they empirically compare variability capabilities in compositional and annotative approaches for SPrL based on EPF Composer and GenArch-P. GenArch-P is a model-driven approach to managing and customizing software process variabilities proposed in (Aleixo et al., 2010). The comparison is based on the same criteria adopted in our study, except that we discarded the uniformity criterion. As in our study, Aleixo et al. come up with better results for the annotative approach.

6 CONCLUSION

This paper presented an empirical qualitative study comparing the representation of variability in compositional and annotative approaches. Such a study provided, although initial, evidence that the annotative approach, in this study represented by SMartySPEM, has more advantages over the compositional approach, represented by EPF Composer. Although the criteria of modularity and detection errors had lower results in the annotative approach, they might be improved by using UML packages for modularity and applying inspection activities for error detection such as in (Geraldi et al., 2015).

As future work, we intend to plan and conduct empirical quantitative studies in order to compare our annotative approach, the SMartySPEM, to other compositional and annotative approaches. In addition, we are working on the establishment of a Scrum-based SPrL by taking real projects experience from industry as well as practitioners as Scrum Masters expertise as there is no real SPrL available in the literature for carrying out empirical studies and evaluating our SPrL-related theories and tools.

As a potential future work, we are considering studying the granularity criterion on the specification of lower-level software process activities, such as, in business process models, allowing one to customize the steps of the activities as, for instance, in multitenancy architectures of Software as a System (SaaS).

ACKNOWLEDGEMENTS

The authors would like to thank Masters and Ph.D. lecturers an practitioners experts for attending this study and for their valuable contribution on assessing the compositional and annotative approaches.

REFERENCES

Aleixo, F., Freire, M., Santos, W., and Kulesza, U. (2010). A Model-driven Approach to Managing and Customizing Software Process Variabilities. In *International Conference on Enterprise Information Systems*, pages 92–100. SCITEPRESS.

Aleixo, F. A., Freire, M., Alencar, D., Campos, E., and Kulesza, U. (2012a). A Comparative Study of Compositional and Annotative Modelling Approaches for Software Process Lines. In *Brazilian Symposium on Software Engineering*, pages 51–60.

Aleixo, F. A., Freire, M. A., Santos, W. C., and Kulesza, U. (2011). Automating the Variability Management, Customization and Deployment of Software Processes: a Model-Driven Approach. In Filipe, J. and Cordeiro, J., editors, *Enterprise Information Systems*, volume 73 of *Lecture Notes in Business Information Processing*, pages 372–387. Springer Berlin Heidelberg.

Aleixo, F. A., Kulesza, U., Freire, M. A., da Costa, D. A., and Neto, E. C. (2012). Modularizing software process lines using model-driven approaches - a comparative study. In *International Conference on Enterprise Information Systems*, pages 120–125. SCITEPRESS.

Aleixo, F. A., Kulesza, U., and OliveiraJr, E. (2013). Modeling Variabilities from Software Process Lines with Compositional and Annotative Techniques: a Quantitative Study. In *International Conference on Product-Focused Software Development and Process Improvement*, pages 153–168.

Behringer, B. (2014). Integrating Approaches for Feature Implementation. In *ACM SIGSOFT International Symposium on Foundations of Software Engineering*, pages 775–778, New York, NY, USA. ACM.

Carvalho, D. D., Chagas, L. F., Lima, A. M., and Reis, C. A. (2014). Software Process Lines: A Systematic Literature Review. In Mitasiunas, A., Rout, T., OConnor, R., and Dorling, A., editors, *Software Process Improvement and Capability Determination*, volume

477 of *Communications in Computer and Information Science*, pages 118–130. Springer International Publishing.

Chemuturi, M. K. and Cagley, T. M. (2010). *Mastering Software Project Management: Best Practices, Tools and Techniques*. J. Ross Publishing, Inc.

Corbin, J. M. and Strauss, A. L. (2008). *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*. Sage Publications, Inc.

Ferreira Filho, J. a. B., Barais, O., Acher, M., Baudry, B., and Le Noir, J. (2013). Generating Counterexamples of Model-based Software Product Lines: An Exploratory Study. In *International Software Product Line Conference*, pages 72–81, New York, NY, USA. ACM.

Galster, M., Weyns, D., Tofan, D., Michalik, B., and Avgeriou, P. (2013). Variability in Software Systems A Systematic Literature Review,. *IEEE Transactions on Software Engineering*, pages 81–90.

Garcia-Borgonon, L., Barcelona, M. A., Garcia-Garcia, J. A., Alba, M., and Escalona, M. J. (2014). Software Process Modeling Languages: a Systematic Literature Review. *Information and Software Technology*, 56(2):103–116.

Geraldi, R. T., OliveiraJr, E., Conte, T. U., and Steinmacher, I. F. (2015). Checklist-based Inspection of SMarty Variability Models: Proposal and Empirical Feasibility Study. In *International Conference on Enterprise Information Systems*, pages 268–275. SCITEPRESS.

ISO (2012). ISO/IEC 15504-5:2012 Information technology – Process Assessment – Part 5: An Exemplar Software Life Cycle Process.

Kalus, G. and Kuhrmann, M. (2013). Criteria for Software Process Tailoring: A Systematic Review. In *International Conference on Software and System Process*, pages 171–180, New York, NY, USA. ACM.

Kästner, C. (2010). *Virtual Separation of Concerns: Toward Preprocessors 2.0*. PhD thesis, Otto von Guericke University Magdeburg.

Kästner, C. and Apel, S. (2008). Integrating Compositional and Annotative Approaches for Product Line Engineering. In *Workshop on Modularization, Composition and Generative Techniques for Product Line Engineering*, pages 35–40.

Kästner, C., Apel, S., and Kuhlemann, M. (2008). Granularity in Software Product Lines. In *International Conference on Software Engineering*, pages 311–320.

Kiczales, G., Lamping, J., Mendhekar, A., Maeda, C., Lopes, C., Loingtier, J.-M., and Irwin, J. (1997). Aspect-Oriented Programming. In *European Conference on Object-Oriented Programming*, pages 220–242. Springer Berlin Heidelberg.

Lee, H. and Kang, K. C. (2013). A Design Feature-based Approach to Deriving Program Code from Features: A Step Towards Feature-oriented Software Development. In *International Workshop on Variability Modelling of Software-intensive Systems*, pages 1–6, New York, NY, USA. ACM.

Linden, F. J. v. d., Schmid, K., and Rommes, E. (2007). *Software Product Lines in Action: The Best Industrial Practice in Product Line Engineering*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Martínez-Ruiz, T., Münch, J., García, F., and Piattini, M. (2012). Requirements and Constructors for Tailoring Software Processes: a Systematic Literature Review. *Software Quality Control*, 20(1):229–260.

OliveiraJr, E., Gimenes, I. M. S., and Maldonado, J. C. (2010). Systematic Management of Variability in UML-based Software Product Lines. *Journal of Universal Computer Science*, 16:2374–2393.

OliveiraJr, E., Pazin, M. G., Gimenes, I. M. S., Kulesza, U., and Aleixo, F. A. (2013). SMartySPEM: a SPEM-based Approach for Variability Management in Software Process Lines. In *International Conference on Product-Focused Software Development and Process Improvement*, pages 169–183. Springer Berlin Heidelberg.

Rombach, D. (2005). Integrated Software Process and Product Lines. In *International Conference on Unifying the Software Process Spectrum*, pages 83–90. Springer-Verlag Berlin.

Sommerville, I. (2015). *Software Engineering*. Pearson, 10 edition.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., and Wesslén, A. (2000). *Experimentation in Software Engineering: an Introduction*. Kluwer Academic Publishers, Norwell, MA, USA.

Becoming Agile in a Non-disruptive Way
Is It Possible?

Ilia Bider and Oscar Söderberg
DSV, Stockholm University, Stockholm, Sweden
ilia@dsv.su.se, oscar.soderberg88@gmail.com

Keywords: Agile, Software Development, Software Engineering, Challenges, Tacit Knowledge, Knowledge Transformation.

Abstract: Due to the increasing popularity of Agile Software Development (ASD), more software development teams are planning to transit to ASD. As ASD substantially differs from the traditional Software Development (TSD), there are a number of issues and challenges that needs to be overcome when transiting to ASD. One of the most difficult challenges here is acquiring an agile "mindset". The question arises whether it is possible to acquire this mindset with the minimum disruption of an already established TSD process. The paper tries to answer this question by developing a non-disruptive method of transition to ASD, while using a knowledge transformation perspective to identify the main features of ASD mindset and how it differs from the one of TSD. To map the current mindset and plan the movement to the mindset that is more agile, the paper suggests using a process modelling technique that considers the development process as a socio-technical system with components that correspond to the phases of the development process. The method suggested in the paper has been designed in connection to a business case of a development team interested to transit to agility in a non-disruptive manner.

1 INTRODUCTION

1.1 Formulating a Problem

Agile Software Development (ASD) has appeared as a reaction on the increasing rate of changes in system requirements, e.g. see (Highsmith et al., 2000): "requirements change at rates that swamp traditional methods". Since 15 years from its inception, ASD from a niched development methodology, mainly used in the web development, made its way to becoming one of the mainstream methodologies. This leads to organizations that use a phase-based methodology become more willing to move to ASD.

Due to the essential differences between the Traditional Software Development (TSD) and ASD, a transition from one to another is quite difficult and includes a number of challenges and pitfalls that are reported in research papers (Conboy et al., 2011; Hajjdiab and Taleb, 2011), books (Smith and Sidky, 2009), and practitioners blogs (Hunt, 2015). The main difficulty here is that an ASD team requires having a "mindset" that differs from the one of a TSD team.

There are a number of books, such as, (Hajjdiab and Taleb, 2011), that suggest methods for transiting from TSD to ASD. However, following these methods presumes that the decision to complete such a transition has been made, and risks attached to the transition understood. In addition, a decision on which brand of Agile, e.g. XP, or SCRUM, to try needs to be taken quite early in the transition process.

Understanding the transition risks and making a right for the given situation choice of the agile practice requires experience. Thus, such a transition has better chances for success if it is led by an experienced person, e.g. an agile coach. Even in this case, there is no guarantee of success. What is more, even if the transition was successful in the end, it could cause a disruption of the existing development process for quite long time. If the existing process does not work, taking the risk and introducing the disruption are fully justified. However, if the process works satisfactory, there could be doubts whether it make sense to jump into the unknown taking the risks and going through the disturbances without knowing whether a better development process will emerge after the transition has been completed.

In connection to the deliberations above, a question arises whether it is possible to gradually transit from TSD to ASD with the minimum

Bider, I. and Söderberg, O.
Becoming Agile in a Non-disruptive Way - Is It Possible?.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 294-305
ISBN: 978-989-758-187-8

disruption of the existing development process? In other words, the question is whether there already exists a method of non-disruptive transition to ASD, and if not, whether such method can be devised. Ideally, such a method should improve the existing development process even before the full transition cycle has been completed. It should be also possible to delay taking the decision on which brand of ASD to use, and even stop the transition at some point being satisfied with what has been achieved, and not taking risks of going farther.

1.2 Overview of a Solution

This paper is a report on the research aimed at answering this question. To the best of our knowledge, there is no non-disruptive method of transition to ASD described in the research or practical literature. Therefore, we use Design Science (DS) approach (Peffers et al., 2007) to answer the question posed above, i.e. we aim to answer it by designing such a method and testing it in practice.

According to the case studies reported in the literature, e.g. (Hajjdiab and Taleb, 2011; Conboy et al., 2011), the biggest issue when transiting to ASD is acquiring the agile mindset by the development team. The latter requires all team to acquire a number of skills, which might not be necessary in the existing TSD. For example, social and communication skills are mandatory for all members, so that they can meet and talk to stakeholders. Therefore, the main focus of our design work is directed to acquiring the agile mindset and a set of skills that is included in it.

To design a method that leads to changing the mindset of the team to the agile mindset, we need to:

1. Find a basis on which to identify the main features of the agile mindset and in what way it differs from the mindset of a more traditional team.
2. Find a way of mapping (modelling) the mindset of the current team so that the difference between the current mindset and the targeted one (agile) can be measured and a plan of action aimed to shorten this distance can be developed.

As far as the first item on the list is concerned, the most commonly used framework for this kind of goal is Agile Manifesto (Agile Alliance, 2001). However, we consider it too vague and allowing multiple interpretations, which leads to misunderstandings and heated arguments in the agile community (Weaver, 2011); see also critique of Agile Manifesto in (Conboy and Fitzgerald, 2004). We needed a more "scientific" basis for developing a non-disruptive method of transition to agile. For this end, we have chosen an approach suggested in (Bider, 2014) that is based on considering TSD and ASD projects from the knowledge transformation perspective. Based on this consideration, (Bider, 2014) defines the essence of ASD in difference from TSD and set some requirements on the structure of the agile project, its team, relations with the customer and techniques used in the project. The results from (Bider, 2014) do not contradict Agile Manifesto, but rather more clearly underline the main features of ASD and the difference between ASD and TSD.

As far as the second item on the list above is concerned, there are a number of methods for evaluating and measuring the current level of agility, see for example (Sidky, 2007). However, mostly, these works rely on Agile Manifesto when determining what the agile mindset is. Furthermore, they are based on the decision of transition to agile being already taken. In addition, these are general methods not connected to the current structure of the development process accepted in the given organization. In other ways, we consider that the existing methods of evaluation of the level of agility do not fit the task of creating a method of non-disruptive transition to agile.

In this work, we have created our own approach to maping (modelling) the mindset of the development team that is suitable for planning steps for advancing the current mindset towards the agile one. This approach is based on the business process modelling technics suggested in (Bider and Perjons, 2015; Bider and Otto, 2015) and called step-relationship modelling in (Bider and Perjons, 2015). The technique uses a system view on the business process considering it as a number of components (or steps) connected with each other via various relationships. The model built according to this technique focuses on depicting these relationships and their properties. When adopting step-relationship modelling technique for our purpose, we concentrated on relationships between the teams that man the components/steps of the given system development process.

One of the main activities in a Design Science (DS) research project is testing the new artefact/solution, which is a method in our case, in at least one real situation. DS does not set a restriction on when in the course of the research project such test needs to be started, e.g. after the design has been finished or in parallel with the design. In our case, the research was conducted in parallel with investigating a business case in the IT department of an insurance company. This department was interested in adopting

a non-disruptive approach of moving towards agility, and it was also used as a test bed for the method. The test is far from being completed, but it was run up-to the department management understood enough of the suggested method and became prepared for completing the first step on the way to agility.

The rest of the paper is structured in the following manner. Section 2 gives a brief overview of the research methodology and knowledge base used in this research and the research background. Section 3 describes the proposed method. Section 4 discusses testing. Finally, in Section 5, we summarize the results achieved and draw plans for the future.

2 RESEARCH BACKGROUND

2.1 The Project History and Methodology

This research has been initiated by the management of an IT department in a large insurance company expressing their interest in transition to a more agile development process. The management did not possess much knowledge on the essence of ASD, or its various brands. They were interested in an approach that included minimum risks and gave a possibility to learn the essence of ASD on the way, while allowing to delay the decision of which particular brand/practice of ASD to adopt. The literary study, part of which is presented in Section 1, has shown that there are a number of practical methods of transition to agile. Nevertheless, none of them was particularly suitable for the requirements that came from the IT department. These requirements were reformulated into the question of "whether it is possible to gradually transit from TSD to ASD with the minimum disruption of the existing development process?" posed in Section 1.1 To answer this question, we decided to develop a "non-disruptive" method of transition to agile.

The development of our method follows the pattern of Design Science (DS) research (Peffers et al., 2007; Baskerville et al., 2009), which is related to finding new solutions for problems known or unknown. To count as a design science solution, it should be of a generic nature, i.e. applicable not only to one unique situation, but to a class of similar situations. DS research can be considered as an activity aimed at generating and testing hypotheses for future adoption by practice (Bider et al., 2013).

Our method development ran in parallel with the investigation of the business case of the IT department in the insurance company. More exactly, we investigated and modelled the structure of the development process in the department including the skill-sets of the process participants and the ways they communicated with each other. The activities were carried out through interviews with representatives of various phases in the process, and studying the internal documentation.

One of the key activity in a DS project is implementation and verification of a generic solution, or artefact in terms of (Peffers et al., 2007), in at least one situation. This activity is also referred to as demonstration or proof of concept in the literature devoted to methodology of DS (Peffers et al., 2007). The demonstration phase in this research is a continuation of our case study. More exactly, we worked out a suggestion on the first steps of the transition to agility for the IT department; and it was accepted by the management. More details on this activity are presented in Section 5.

As already has been mentioned in Section 1, we used some existing theoretical frameworks as a knowledge base when developing our method. As we do not expect that these frameworks are known to the reader, in the next sub-sections, we give a short overview of them before presenting our method.

2.2 Agility from the Knowledge Transformation Perspective

In this section, we give a short summary of TSD and ASD models built based on the knowledge transformation perspective presented in (Bider, 2014). These models, in their own turn, are built based on the SECI model (Nonaka, 1994). SECI stays for Socialization – Externalization – Combination – Internalization, and it explains the ways of how knowledge is created in an organization while being transformed from the tacit form (in the heads of the people) to the explicit one (e.g. on the paper) and back, see Figure 1. The cycle of knowledge creation consists of the following four steps or phases:

1. The cycle starts with *Socialization*, where tacit knowledge is transferred from the heads of one group of people to others via informal means, such as conversations during the coffee breaks, meetings, observations, working together, etc.
2. The next phase is *Externalization*, which is the conversion of knowledge from the tacit form into the explicit one, e.g. a model of situation.
3. The third phase is *Combination*, which is transforming the externalized (explicit) knowledge in a new form using existing knowledge, e.g. solution design principles.

4. The last phase is *Internalization*, which is converting the explicit knowledge, e.g. a solution, in the tacit knowledge of people who will apply this knowledge to any situation that warrants it.

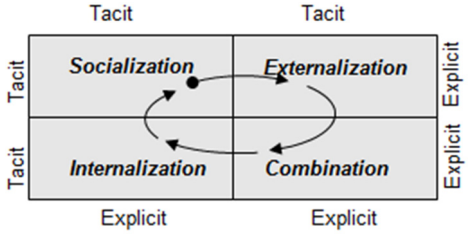


Figure 1: SECI diagram of knowledge creation.

Applying ideas from SECI to software development, (Bider, 2014) designed two models of knowledge transformation in software development projects, one - for Traditional Software Development (TSD), and another - for Agile Software Development (ASD). Both are presented in Figure 2. In both cases, the knowledge transformation cycles starts with tacit knowledge possessed by stakeholders on problems/needs to be solved/satisfied by a new software system. The next step common for both models is embedment when the knowledge on a solution becomes embedded in the system that is considered by its users as a whole possessing its own behaviour. The last step in the knowledge transformation in both models is adoption – transforming the knowledge embedded in the system into the tacit knowledge of the system's users on how to use this system in various working situations.

The models for TSD and ASD in Figure 2 substantially differ in the following aspects:

1. The nature of the first phase in ASD differs from that of TSD. It consists in transferring tacit knowledge on the problem and needs from the stakeholders to the development team. This phase corresponds to *Socialization* in Figure 2. Also, *Design* and *Coding* are merged into one phase Embedment. This can be defined as the first motto of agility: "Avoid or delay explication of knowledge as much as possible. Ideally go from tacit knowledge directly to the embedded one."
2. In addition, one big cycle is substituted by many smaller and shorter ones. The system is built iteratively starting with the basic functionality. During the exploitation of the basic system, better understanding of the needs is acquired, which is converted in adding details to the system in the next iterations. In other words, the second motto of agility can be defined as: "Develop and introduce in practice as little as possible as soon as possible, and build upon it in the following iterations".

Based on the analysis of the knowledge transformation models for TSD and ASD, (Bider, 2014) identifies 6 properties of the development process that differentiate TSD and ASD; these are presented in Table 1. The first three properties, *team*, *user involvement* and *agreement*, belong to the social

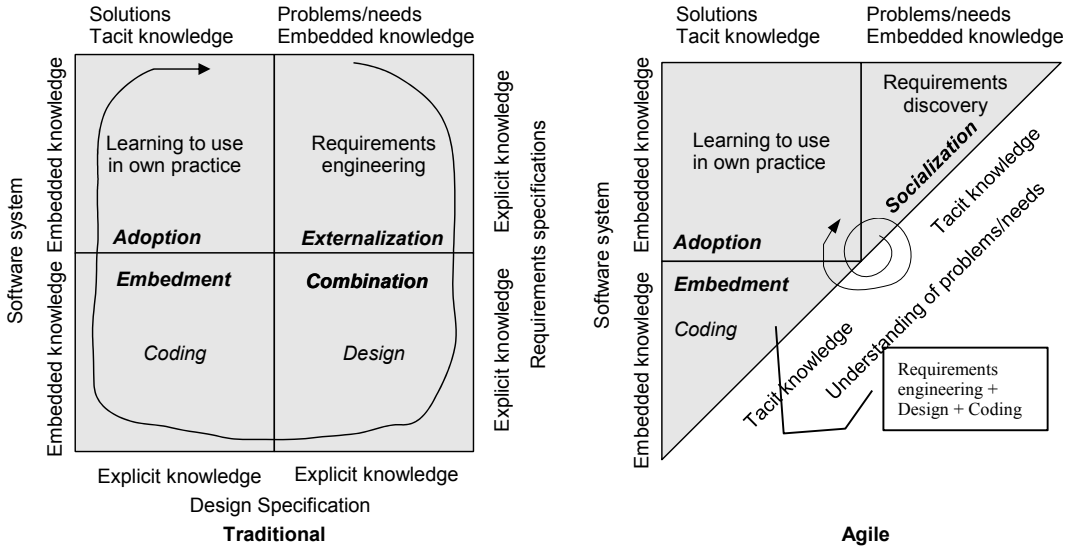


Figure 2: Left – ECEA model (Externalization-Combination-Embedment-Adoption) for TSD. Right - SEA model (Socialization-Embedment-Adoption) for ASD. Adapted from (Bider, 2014).

perspective of system development, while the second three properties, *core system*, *architecture* and *tools*, belong to the technical perspective of system development. We will be using these differentiating properties when developing our non-disruptive method later in Section 4.

Table 1: Properties that differentiate ASD from TSD.

#	ASD	TSD
1	One *team* consisting of "universal" members	Several specialized *teams*
2	*Stakeholders involvement* during the duration of the project	*Stakeholders involvement* during the *Externalization* and *Adoption* phases
3	Non-contractual *agreement* based on trust	*Contractual* agreement is possible
4	Possibility to identify and agree on a *core system* that can be expanded in consequent iterations	Not mandatory, but can be employed.
5	*Architecture* aimed at expansion	*Architecture* aimed at fulfilling the identified requirements
6	Employing *high-level tools*, e.g. domain-specific languages, development platforms, libraries	Not mandatory – low level, and universal tools can be employed

2.3 Step-relationship Model

A step-relationship model represents a business process as a (relatively) small number of steps (Bider and Perjons, 2015), or functional components (Bider and Otto, 2015), connected with each other through various types of relationships. Each type of relationships, i.e. a relation in a mathematical sense, represents a separate view of the model.

There are two ways of representing a relationships type, graphical and matrix. In the graphical form, the steps/components are presented as rectangles (boxes), while arrows between the rectangles show relationships between the corresponding steps/functional components. Labels inside the rectangles name the steps, while labels on the arrows give additional characteristics to the relationships. As an example, Figure 3 represents output-input relationships in a sample software development process. Each arrow shows formalized output of one step/component serving as an input to another step/component.

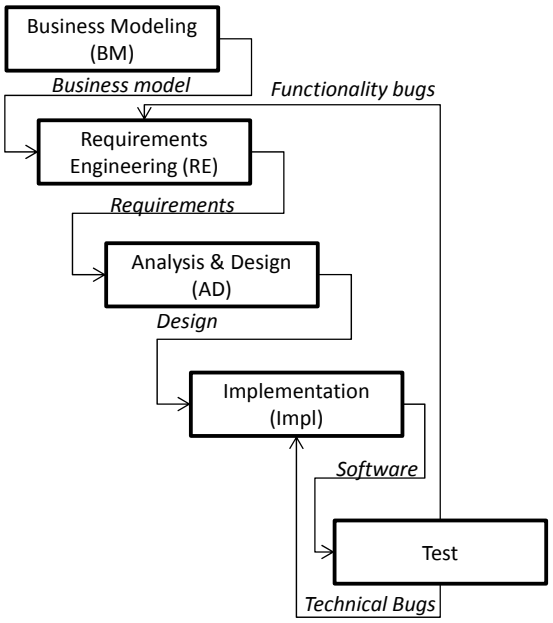


Figure 3: Graphical presentation of relationships.

In the matrix form, a relationships type is represented as a square matrix where both columns and rows correspond to steps/components of the process. A sell (*a*,*b*) where *a* is a column and *b* is a row is reserved for describing a relationship of the given type between step *a* and step *b*, if any exists. As an example, Table 2 presents the same output-input relationships type as Figure 3, but in the matrix form. More examples of relationships in the graphical and matrix forms are presented in Section 3.

Table 2: An example of presenting relationships in the matrix form.

	BM	RE	AD	Impl	Test
BM					
RE	Model				Bugs
AD		Reqs			
Impl			Design		Bugs
Test				Software	

3 DESIGNING A METHOD

3.1 Creating a Single Team

There are several essential properties of ASD that need to be achieved in order to successfully transit to agile. When developing our method, we assume that at least some of them can be achieved without essentially changing the current process. We also

assume that it is possible to somehow measure the progress achieved on the way.

According to the first row in Table 1, ASD has a single development team of members that could do all kind of work in the process, including talking to the stakeholders and programming. This is not mandatory for TSD, where separate specialized nonintersecting teams can complete the job. Also, in a single ASD team, all members communicate with each other frequently, which is not required in TSD. In TSD, informal communication in the frame of the development process may concentrate inside each specialized team, while the formal output-input channels are used for passing over the job between the teams, as is represented in Figure 3, and Table 2.

The two properties of (a) having specialized teams and (b) lack of communication between the teams are related to each other. A narrow specialization may create a hinder for communication due to differences in professional jargons and culture.

Based on the deliberation above, we have identified two properties of the development process that need to be measured and improved, in the first hand, when transiting to the agile approach. These are: (a) intensity of communication between the teams, and (b) ability of members of one specialized team to do the job assigned to the other teams. These two properties can be represented via relationships between the teams manning the steps. Technically, these relationships can be represented with the help of two matrixes: (a) the *communication intensity matrix*, and (b) the *cross-competency matrix*, as is discussed in the next subsections.

3.1.1 Increasing Communication Intensity

An example of the *communication intensity matrix* for the model in Figure 1 is presented in Table 3. A cell (*a,b*) in the communication intensity matrix, where *a* stays for a column and *b* for a row, defines the intensity of communication between teams of steps *a* and *b* initiated by team *a*. Interpretation of the values in the cells depends on the level of separation between the teams, e.g. one site or multiple sites. In the example presented in Table 3, communication are supposed to take place in the form of meetings, were *High* means daily communications meetings. *Average* means 3 times a week, *Low* means once a week. Empty cells outside the diagonal mean that no communication happens between the corresponding teams.

Note that the communication intensity matrix is aimed at characterizing the intensity of communication between the specialized teams, assumption being that inside the teams their members communicate/collaborate in a natural way. If this is not true, the diagonal of the matrix can be used for representing communication intensity inside the teams.

Table 3: An example of a communication intensity matrix.

	BM	RE	AD	Impl	Test
BM		High	Average		Low
RE	High		Average	High	Low
AD	High	High		High	
Impl	Low		Average		High
Test	Low	Low	Average	High	

The communication intensity matrix can be used for both depicting the communication intensity in the current state and planning for increasing the communication intensity. The latter can be done by changing values of some cells in the matrix to reflect the goal of increasing communication intensity. To facilitate the planning work, we have transferred some information from the output-input matrix, see Table 2, to the communication intensity matrix in Figure 3. More specifically, we make the borders of cell (*a,b*) thick in all cases where cell (*a,b*) is not empty in the output-input matrix. The latter means that the column step *a produces* a formalized input for the row step *b*, e.g. design specification. In addition, we made the background of cell (*a,b*) grey in case cell (*b,a*) is nonempty in the output-input matrix (Table 2). The later means that the column step *b receives* formalized output from the row step *b*.

Formally, the result of adding thick borders and grey background means that the matrix presented in Table 3 is a merger of a "pure" intensity communication matrix (without thick borders and grey background) with the simplified output/input matrix (the content of the cells in the latter is not represented in the merger) and a transposition of the latter. The merged communication extensity matrix is more convenient for planning the next step of transition to agile as described below.

One can expect that communication should be more extensive between the steps that are connected with an output-input relationship. Formalized outputs, like requirements or a design specification, in a software development process cannot be made totally formal, and they need interpretation from the receiving team. Misinterpretation can lead to a wrong system being delivered to the customer. The thick border represents the needs of informal explanation of the formalized output when it is being transferred to the receiving team. The grey background represents the need for communication between the

receiving team and the producing team while the former is doing their part of work. Even when the receiving team get the informal explanations on their formalized input, there can be a need to verify their understanding from the originator of the input. For example, the designers may need to contact the requirements engineers later on when they start converting certain requirements into design. In (Bider and Perjons, 2015), this type of backward communication is called week dependencies, while (Bider and Otto, 2015) refer to them as to feedback links.

Summarizing the above, when planning the next goal in intensifying the communication between the teams, it is worthwhile to start intensification that corresponds to cells with thick borders or grey background. For example, the next goal for the situation presented in Table 3, could be the one described in Table 4, where the difference is presented in bold. The difference consists of intensifying forward communication between *Analysis & Design* and *Implementation*, and backward communication between *Analysis & Design* and *Requirements Engineering*. Such measure makes sense even for improving the already existing process.

Table 4: Next step in communication intensity.

	BM	RE	AD	Impl	Test
BM		High	Average		Low
RE	High		**High**	High	Low
AD	High	High		High	
Impl	Low		**High**		High
Test	Low	Low	Average	High	

3.1.2 Increasing Cross-Competency

While the communication intensity matrix can be considered as a tool of intensifying internal communication in the future single team, the cross-competency matrix can be considered as a tool for achieving "universality" of its members (see the first row in Table 1). An example of such a matrix is presented in Table 5. In this matrix a cell (*a,b*), where *a* stays for a column and *b* for a row, defines the percentage of the team *a* members that have working knowledge on the tasks completed in the step *b*. An empty non-diagonal cell means 0%. Here, having working knowledge on a specific task means that a person in question has some practical experience of this task.

As with the communication intensity matrix, we add to this matrix some information from the output-input matrix in the form of thick borders around cells and grey background. This information is aimed at helping to plan the next step of transition to agile. Marked cells should be targeted for increasing cross-competence in the first place, as this can decrease the risks of misinterpretation of the formalized inputs and misunderstanding in communications. Such measure might be helpful even for improving the existing process.

Table 5: An example of cross-competency matrix.

	BM	RE	AD	Impl.	Test
BM		50%	75%		
RE	75%		75%		50%
AD	75%				
Impl.	50%	50%	75%		50%
Test	50%				

An example of the next planned step for the situation presented in Table 5 is presented in Table 6, where the difference is presented in bold. The difference consists of increasing cross-competency of the *Requirements Engineering* and *Implementation* Teams.

Table 6: next step in cross-competency.

	BM	RE	AD	Impl	Test
BM		50%	75%		
RE	75%		75%		50%
AD	75%	**50%**		**50%**	
Impl	50%	50%	75%		50%
Test	50%			**50%**	

As cross-competency requires *working* knowledge of the tasks completed by other teams, it is not enough just to send people to a course. The proper way of achieving cross-competency in cell (*a,b*) in the frame of the existing software development process is to send some people from team *a* to work in team *b* for some time. This can degrade the overall performance in the beginning, but this one-time cost is worth taking, as increase in cross-competency minimizes the risk of producing the wrong software (see deliberation above).

When planning increase in cross-competency for *Implementation* step with other teams, it is worthwhile to consider row 6 in Table 1 that refer to using high-level tools. This property has not been introduced for the sake of creating a single team of "universal" members, but for being able to complete development loops in a speedy manner. However, having high-level development tools may also help in acquiring programming skills by people without

technical education. So, if such tools are not already employed, it could be advantageous to start transition from low-level programming to using high-level tools before increasing competency in programming in other teams.

3.2 Avoiding Explication of Knowledge

As was discussed in Section 2.2, one of the ASD principles is to delay or avoid explication of knowledge, ideally, by going from the tacit understanding of problems/needs to building software. This implies skipping creating detailed requirements and design specifications. More specifically, requirements are left on the tacit level as a general understanding/image of the problems and needs, while design is done via proper structuring of the code. The latter could be facilitated by using high-level development tools, like domain specific languages, component libraries.

Avoiding explicit requirements and design does not mean that these activities are excluded; they are done on the tacit level. To reach the level of proficiency when requirements and design are done on the tacit level is difficult, if ever possible, without obtaining skills in both requirements engineering and design. Obtaining these skills by all team members in the frame of the existing phase-based process has already been discussed in Section 3.1.

The next question is how to shorten the time period from the first contact with the customer to starting producing executable code while still remaining in the frame of a traditional software development project. We believe that this can be achieved by gradual transition from sequential execution of the steps of development process to the semi-parallel execution. The latter means starting the design before all requirements are discovered, and starting coding before all design specifications are created.

The current level of parallelism can be represented in a graphical form as a timeline intensity diagram (Bider and Otto, 2015). An example of such a diagram that corresponds to Figure 3 is presented in Figure 4. The difference between Figure 3 and 4 is that in Figure 4, the shapes representing steps do not have rectangular form. The upper border of the shape can be of any form representing the increase/decrease in the amount of work being done at certain moments of time. The intensity of work can be increasing or decreasing with time, or can be first increasing and then decreasing or vice versa (not illustrated in Figure 4). In addition, the step shapes in Figure 4 are placed in the order they are executed. If some steps run partly in parallel, the projections of their shapes on the time axes will intersect. In the example of Figure 4, there are two occasions of the parallelism, namely (1) step *Analysis & Design* runs partly in parallel with *Implementation*, and step *Implementation* runs partly in parallel with *Test*.

Timeline intensity diagram can be used for planning the next goal for transition to agile in the same way as communication intensity and cross-competency matrices are used, see Fig. 5.

In the example of Figure 5, all steps run partially in parallel, which is rather a radical change when starting from Figure 4. If such a transition is too difficult to complete in one go, then smaller goals can

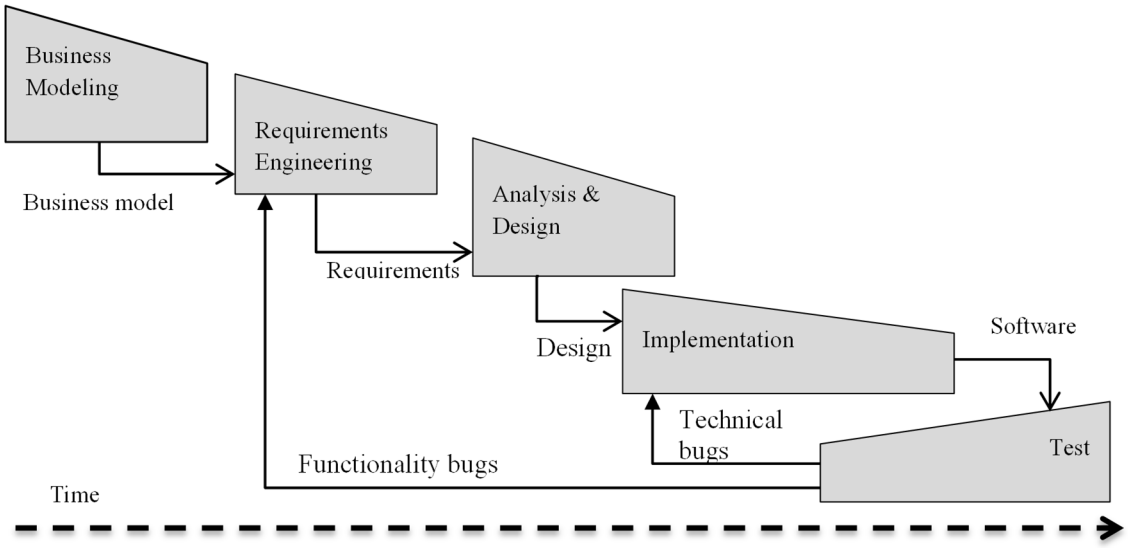


Figure 4: An example of timeline intensity diagram.

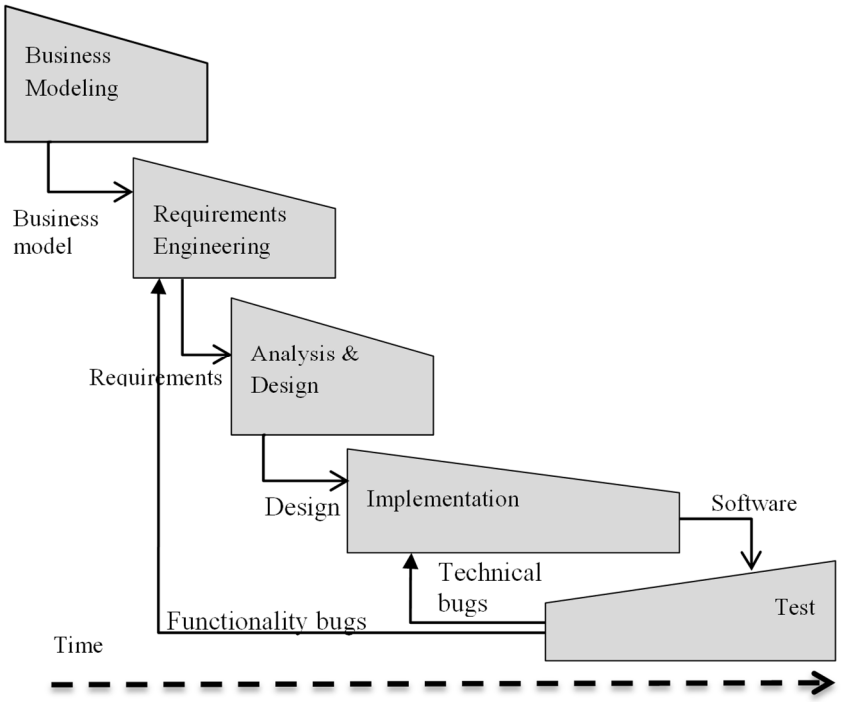


Figure 5: An example of timeline intensity diagram to be achieved.

be set in between, e.g. where only two new steps run in parallel.

Working in parallel means that the formalized output is delivered to the next step in portions. This requires understanding of how the formalized output is used by the next step so that each portion is relatively independent and can be successfully used by the team of the next step for producing its own formalized output. Thus parallel execution requires certain degree of cross-competency on behave of the output producer. In addition, it requires efficient communication channels between the steps. Parallel execution of steps in software development bares a risk that the already produced portion of the given step output, e.g. requirements, can be negated when the work progresses. If this "negated" portion has already been sent to the next step, e.g. design, and is under processing of this step's team, then the information on the negation should be immediately made available for this team. Getting this information can stop or postpone their activities related to the questionable portion of the requirements. Note that with an experienced team, the advantages of running in parallel, e.g. shorten time, overweight the risks described above.

Summarizing the deliberation above, transition to parallel execution of two steps should be planned when a certain degree of cross-competency and communication intensity between these steps has already been achieved.

It is also worthwhile to mention that portioning of the output needs to take into account architectural considerations. Portions that are sent first need to be significant for building a skeleton of the architecture, and portions that are sent later should be relatively independent of each other and should not considerably affect the architecture.

3.3 Other Considerations

In the previous part of this section we mainly discussed three issues that can help in transition to agile: inter- step communication, cross-competency and parallel execution. Furthermore, we touched the issue of high-level development tools that facilitates both achieving cross-competency, and excluding explicit design. In addition, we also touched the architectural issues that need to be taken care of when planning transition to the parallel execution of steps. We also have shown that all these issues are interconnected and should be considered together when planning transition to agile.

We believe that after dealing with the issues discussed in this section the team will acquire the agile mindset, and became prepared for sorting out the remaining issue on the way to agility. Consider, for example, the issue of stakeholders involvement during the whole project. Such involvement is

impossible to arrange in a traditional phase based development process based on two reasons. Firstly, people outside business modelling and requirements engineers might not have competency of talking to non-technical people. Secondly, non-technical stakeholders seldom understand technical documentation, which will prevent their engagement. The first problem can be solved through cross-competency, and the second - through parallel execution that ensures that the new portion of software will be produced in a speedy fashion, and could be demonstrated and discussed with the stakeholders.

4 TESTING THE METHOD

As has already been discussed in Section 2.1, development of our non-disruptive method of transition to agile was done in parallel with a case study in the IT department of a large insurance company. The first phase of the study was connected to the development of the method, and the second phase with testing it.

The first phase was completed based on the internal process documentation and interviews with representatives of different teams engaged in the development process. Based on the information obtained, it was decided that the three most important aspects that need to be mapped when describing the current state of affairs were communication intensity, cross-competency and timeline intensity. The step relationship modelling technique (Bider and Perjons, 2015; Bider and Otto, 2015) was chosen for representing these aspects. The concept of the timeline diagram was already known from (Bider and Otto, 2015), while the communication intensity matrix and cross-competency matrix where designed during the current project.

Based on the internal documentation and information from the interviews, a model of the current development process was produced. This model is closed to the one presented in Figure 3 and 4, and Tables 2, 3 and 5, except that one step from the original model is omitted. The structure of the communication intensity and cross-competency matrixes in the original model were somewhat simpler than what was presented in Tables 3 and 5. More exactly, the details that came from merging with the output-input matrix were absent; they were added when we worked on this paper.

The test phase of the case study consisted of: (a) suggesting the next desired state of the development project, which roughly corresponds to the one presented in Tables 4 and 5 and Figure 5, and (b) presenting the suggestions to the IT department management. The goal of the test phase was twofold, namely, to check

1. Whether the method could be understood by people not very familiar with the agile practices.
2. Whether they can accept concrete suggestions based on this method, provided that they are approved by the higher management. This check (approximately) corresponds to "reediness to use" in Technology Acceptance Model (Davis, 1989).

The check has been completed by presenting the method and an action plan based on this method to the management of IT-department that consisted of 4 persons. After the presentation, an interview has been conducted with each person based on the following 4 questions/topics:

1. Based on the presentation, have you understood what kind of organizational changes the transition to agile will require?
2. Based on the presentation, have you understood the action plan for movement towards a more agile development process?
3. Based on the presentation, are you prepared to submit the action plan to the upper/higher management for approval?
4. Based on the presentation, are you prepared to set the suggested plan in action if approved by the higher management?

For the questions 1, 2 and 4 the answers were on the positive side from all respondents. When answering question 3, some respondents expressed doubts whether just presenting the action plan to the higher management is enough to influence the approval. However, all of them agreed that such a presentation makes sense. The doubts on influencing the decisions were connected to the plan itself not explaining the benefits to be obtained. However, another opinion was that presenting the action plan could initiate discussions that would lead to understanding the benefits. Anyway, the discussion around the third topic explicated the needs to explain the benefits achieved even before the full transition to agile has been completed. This served us as a motivation to insert the discussion on such benefits in various places of this paper.

Summarizing the lessons learned about our non-disruptive method of transition to agile from the case study, we can state that:

1. It is possible to model the current state of the

development process and suggest a plan of actions for transition to agile.

2. The method is understandable for the professionals in software development not familiar with the details of the agile practices. What is more, the plan of actions based on the method is considered to be "doable", and could be accepted for implementation, provided the approval of the higher management is obtained. Though, there are some doubts that such approval is easy to obtain, presenting the plan of action to the higher management could initiate a discussion that could lead to its acceptance.

The lessons above were obtained based only on one case study. However, from our practical experience, the IT department in the study is just an ordinary system development organization, and there is no reason to suggest that the lessons learned will substantially differ when the method is applied to another organization of the same kind.

In short, we consider the check for "readiness to use" as completed with positive results. On its own, such a check does not guarantee that an organization can actually execute a plan of action developed based on the method. However, we consider this check encouraging enough for continuing the efforts of further development and testing the method.

5 CONCLUSIONS

There are ample evidence, provided in the literature referenced to in Section 1, of existence of challenges and difficulties when completing a transition from TSD to ASD. These can be attributed to such a transition being a major organizational change for a software development organization, and it is well known that any organizational change is difficult to complete due to an organization, as a system, always resists any change.

According to (Regev, 2015), the best prerequisite for successful organizational change is stability. Therefore, a system development organization with a well-functioning TSD process does not need to "jump" on a radical pass to ASD, but should consider using the existing process as a tool for successful transition to ASD. The non-disruptive method of transition to ASD described in this paper gives an example, of how an organization can practically conduct the transition via using the existing process as a tool.

Summarizing the results achieved so far, we can identify three major contribution of this work:

1. To the best of our knowledge, the contemporary literature does not have an explicit definition of a goal of using the existing development process as a platform/tool for transiting to agile. Therefore, our explicit formulation of this goal constitutes the first contribution of this paper. This contribution appears in the title and is discussed in more details in Section 1.
2. In Section 3, we have introduced three types of measurements that can be used to determine the level of agility achieved while the organization is still following TSD: communication intensity, cross-competency, and the level of parallelism. This are easy to understand measures, and as our test case shows can be obtain through interviewing people working in the project. This measures can be used independently whether the organization wants to transit to agile in disruptive or non-disruptive manner.
3. Lastly, this paper also contains a draft of the non-disruptive method of transition to agile that has gone through the initial test of designing a plan of actions and acquiring "readiness to use" in a typical software organization. More tests and further development are required to confirm the validity of the method. However, the work done so far (including the initial test) is sufficient to show that, at least theoretically, a non-disruptive method for transition to agility can be built. Publishing this work might inspire other researchers and practitioners to seek own ways for a non-disruptive transition.

One difference of our method of transition to agile with those of other (some of them are referred to in Section 1) is that we pursue a special goal of using the current development process as a tool/platform for the transition. Another difference is the theoretical basis on which the method has been built. Normally, other researchers and practitioners use Agile Manifesto (Agile Alliance, 2001) as a basis for building a method. Instead, we use the theoretical underpinning of agility based on the knowledge transformation perspective from (Bider, 2014). This perspective has helped us to choose the most important issues on which to focus when transiting to agile. What is more, the issues, when resolved, may improve the current development process even before the full transition can be completed.

Our plans for the future include further development and testing of the non-disruptive method, as well as dissemination of results, especially

among practitioners. The latter activity is considered as an important one in the Design Science research (Peffers et al., 2007). The reason for its importance is that the researchers themselves have no possibility to fully test a new design, aside of conducting demonstration in few cases. The real test can be completed only when (and if) the industry adopts the method so that more test cases become available for study.

ACKNOWLEDGEMENTS

The authors are in debts to the management and developers of the IT department of the insurance company who initiated this research and spent their time answering interview questions, and listening, discussing and accepting our suggestions.

REFERENCES

Agile Alliance, 2001. *Manifesto for Agile Software Development*. (Online) Available at: http:// agilemanifesto.org (Accessed 10 October 2013).

Baskerville, R.L., Pries-Heje, J. & Venable, J., 2009. Soft Design Science Methodology. In *DERIST 2009*. ACM, pp.1-11.

Bider, I., 2014. Analysis of Agile Software Development from the Knowledge Transformation Perspective. In Johansson, B., ed. *13th International Conference on Perspectives in Business Informatics Research (BIR 2014)*. Lund, Sweden. Springer, LNBIP 194, pp.143-57.

Bider, I., Johannesson, P. & Perjons, E., 2013. Design science research as movement between individual and generic situation-problem-solution spaces. In Baskerville, R., De Marco, M. & and Spagnoletti, P. *Organizational Systems. An Interdisciplinary Discourse*. Springer. pp.35-61.

Bider, I. & Otto, H., 2015. Modeling a Global Software Development Project as a Complex Socio-Technical System to Facilitate Risk Management and Improve the Project Structure. In *Proceedings of the 10th IEEE International Conference on Global Software Engineering (ICGSE), forthcoming*. Ciudad Real, Spain. IEEE.

Bider, I. & Perjons, E., 2015. Design science in action: developing a modeling technique for eliciting requirements on business process management (BPM) tools. *Software & Systems Modeling*, 14(3), pp.1159-88.

Conboy, K., Coyle, S., Wang, X. & Pikkarainen, M., 2011. People over Process: Key Challenges in Agile Development. *IEEE Software*, 28(4), pp.48-57.

Conboy, K. & Fitzgerald, B., 2004. Toward a conceptual framework of agile methods: a study of agility in different disciplines. In *Proceedings of the 2004 ACM workshop on Interdisciplinary software engineering research*. Newport Beach. ACM, pp.37-44.

Davis, F.D., 1989. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly*, 13(3), pp.319–40.

Hajjdiab, H. & Taleb, A., 2011. Adopting Agile Software Development: Issues and Challenges. *IJMVSC*, 2(3), pp.1-10.

Highsmith, J., Orr, K. & Cockburn, A., 2000. *E-Business Application Delivery, pp. 4-17*. (Online) Available at: www.cutter.com/freestuff/ead0002.pdf .

Hunt, A., 2015. *The Failure of Agile*. (Online) Available at: http://blog.toolshed.com/2015/05/the-failure-of-agile.html (Accessed October 2015).

Nonaka, I., 1994. A dynamic theory of organizational knowledge creation. *Organ. Sci.*, 5(1), pp.14–37.

Peffers, K., Tuunanen, T., Rothenberger, M.A. & Chatterjee, S., 2007. A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, 24(3), pp.45-78.

Regev, G., 2015. *Fundamental Systems Thinking Concepts for IS Engineering: Balancing between Change and Non-change*. (Online) Stockholm University Available at: http://sched.co/2OGV (Accessed October 2015).

Sidky, A., 2007. *A structured Approach to Adopting Agile Practices: The Agile Adoption Framework. PhD Thesis*. (Online) VirginiaTech Available at: http://scholar.lib.vt.edu/theses/available/etd-05252007-110748/ (Accessed October 2015).

Smith, G. & Sidky, A., 2009. *Becoming Agile*. Greenwich, CT: Manning.

Weaver, M., 2011. *Do you agree or disagree that Scrum is not Agile?* (Online) Available at: http:// www.linkedin.com/groups/Do-you-agree-disagree-that-81780.S.52354777 (Accessed April 2014).

Knowledge Mapping in a Research and Development Group
A Pilot Study

Erivan Souza da Silva Filho, Davi Viana, Jacilane Rabelo and Tayana Conte
USES Research Group, Instituto de Computação, Universidade Federal do Amazonas, Manaus, Brazil
{essf, davi.viana, jaci.rabelo, tayana}@icomp.ufam.edu.br

Keywords: Knowledge Management, Knowledge Mapping, Knowledge Map.

Abstract: In Enterprise Systems, representing the flow of knowledge may indicate how participants work using their knowledge. Such representation allows the understanding of how knowledge circulates between the development team and improvement opportunities. Knowledge Management supports the management of knowledge through techniques that identify how knowledge behaves in projects. One of these techniques is Knowledge Mapping, which supports representing how participants share their knowledge, which sources of knowledge are consulted and which people it helps during a project. However, to draw up a knowledge map, we need a process for capturing and analyzing data that can extract information that reflect these aspects. This work aims at presenting a process for Knowledge Mapping to develop a map indicating what knowledge the participants used, who or what they accessed and indications of its core competencies. Additionally, this paper discusses a pilot study regarding the application of the proposed process. As a result, we generated a knowledge map for a software engineering research and development group, in which contains a set of profiles and features what the main skills that a participant uses are.

1 INTRODUCTION

The main asset of Software Companies is knowledge. Thus, it is necessary to manage this knowledge and use their experiences in development activities (Hansen and Kautz, 2004). In any industrial or academic environment, there are people who have knowledge, and it may be of interest to promote such knowledge management (Krbálek and Vacek, 2011).

Knowledge management is the process of creating, validating, representing, distributing and applying knowledge (Bhatt, 2001). Knowledge management also refers to identifying and increasing the collective knowledge in an organization to help it become more competitive (Alavi and Leidner, 2001).

The goal of these efforts is to provide members of the organization with the knowledge they need to maximize their effectiveness, thus improving the efficiency of the organization (Mitchell and Seaman, 2011). The environment or territory in the context of knowledge management is not geographical, but intellectual (Eppler, 2001), where we need techniques that seek to represent the main aspects of that environment.

One of the techniques in Knowledge Management that seeks to represent these aspects is Knowledge Mapping. Knowledge mapping is a process of surveying, assessing and linking the information, knowledge, competencies and proficiencies held by individuals and groups within an organization (Anandarajan and Akhilesh, 2012).

The result of a mapping is a Knowledge Map that shows the relationships among the procedures, concepts and skills, which provides easy and effective access to sources of knowledge (Balaid *et al*., 2013). The main purpose and benefit of a knowledge map are to show people from within the company where to go when they need knowledge (Davenport and Prusak, 1998).

This paper presents a process of knowledge mapping that aims at representing the flow of the employees' knowledge within software organizations. We combined some approaches in order to create such process. This paper also describes the results of a pilot study in which the proposed process was applied in a Research and Development (R&D) group.

The remainder of this paper is organized as follows. Section 2 presents our theoretical reference. Section 3 presents the developed knowledge mapping process. Section 4 shows planning process

Filho, E., Viana, D., Rabelo, J. and Conte, T.
Knowledge Mapping in a Research and Development Group - A Pilot Study.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 306-317
ISBN: 978-989-758-187-8

of the pilot study. Section 5 discusses the results obtained in the pilot study. Finally, Section 6 presents our conclusions and future work.

2 THEORETICAL REFERENCE

Individual knowledge is necessary for the development of knowledge within an organization (Bhatt, 2001). Knowledge within an organization is a collection of knowledge, experiences and information which people or groups employ to carry out their tasks (Vasconcelos *et al*., 2005). This section shows the theoretical reference and the main concepts for this work.

2.1 Knowledge Management

Human resources are the main assets of many companies where knowledge has to be preserved and passed from the individual to the organizational level, enabling continuous improvement and learning (Lindvall *et al*., 2003). Companies generally understand Knowledge as how information is encoded with a high proportion of human value-added, including perception, interpretation, context, experience, wisdom, and so on (Davenport and Völpel, 2001).

Davenport and Prusak (1998) made a distinction between data and information. Data is a group of distinct facts and goals related to events. Information aims at changing the way in which the receiver perceives something, exercising some impact on his/her judgment and behavior.

Nonaka and Takeuchi (1995) states that knowledge, unlike information, is about beliefs and commitment, and characterize it into two types: explicit and tacit. Explicit or codified knowledge can be articulated in formal or textual language. Tacit knowledge is the personal knowledge, incorporated to the individual experience, and that involves intangible factors (e.g. personal beliefs, perspectives and value systems).

Knowledge Management is a method that simplifies the process of sharing, distributing, creating and comprehending a company's knowledge (Bjørnson and Dingsøyr, 2008). Its goal is to solve problems regarding the identification, localization and usage of knowledge (Rus and Lindvall, 2002).

A prerequisite for the strengthening of knowledge management is a good understanding of how knowledge flows within the organization (Hansen and Kautz, 2004). The identification of the knowledge flow shows us the way on which new concepts and ideas are spread, which can be useful to facilitate changes in management initiatives (Gourova *et al*., 2012). One of the applied techniques for searching and defining organizational knowledge flow is knowledge mapping.

2.2 Knowledge Mapping

Knowledge mapping is a process, method, or tool made for analyzing knowledge in order to discover characteristics or meanings, and view knowledge in a comprehensible and transparent manner (Jafari *et al.*, 2009). The purpose of knowledge mapping is to seek a better orientation in a given domain and access knowledge from the right people at the right time (Krbálek and Vacek, 2011).

One of the advantages of knowledge mapping includes the freedom to organize without restriction, meaning that there are no limits to the number of ideas and connections that can be made (Nada *et al.*, 2009). Knowledge mapping usually takes part of Knowledge Audit processes and methodologies.

Elias *et al*. (2010) define Knowledge Audit (KA) as the identification, analysis and evaluation of the activities, processes and practices for managing the knowledge that a company already has.

Knowledge Audit is used to provide an investigation into the organization's knowledge about the health of knowledge (Elias *et al*., 2010), identifying and understanding the knowledge needs in organizational processes.

Meanwhile, by using Knowledge Mapping techniques would show a logical structure of relationships between tacit human knowledge and explicit knowledge in documents (Krbálek and Vacek, 2011). The result of knowledge mapping is a knowledge map.

2.3 Knowledge Map

Knowledge Map is a diagram that can represent words, ideas, tasks, or other items linked to and arranged in radial order around a central key word or idea (Nada *et al.*, 2009). Furthermore, it is an interactive and open representation that organizes and builds structures and procedural knowledge used in the pursuit of exploration and problem solving (Anandarajan and Akhilesh, 2012).

Knowledge maps also provide a holistic view of knowledge resources (Balaid *et al*., 2013). Eppler (2001) distinguishes five types of Knowledge Maps, shown in Table 1. The five maps can be combined to generate new mapping techniques.

Table 1: Types of Knowledge Maps (Eppler, 2001).

Name	Description
Knowledge Source Maps	These are maps that structure a population of experts from a company through search criteria, such as their knowledge domain, proximity, length of service or geographical distribution.
Knowledge Asset Maps	This type of map visually describes the storage of knowledge of a person, a group, a unit or an organization.
Knowledge Structure Maps	It is the overall architecture of a knowledge domain and shows how parts relate to each other. It assists managers in understanding and interpreting a specialized field.
Knowledge Application Maps	It shows what kind of knowledge must be applied at certain stages of the design process or in a specific business situation. It answers the question of which people are involved in an intensive knowledge process, such as auditing, consulting, research or product development.
Knowledge Development Maps	These maps can serve as development pathways or visual learning which provide a common corporate vision for organizational learning.

2.4 Related Work

There are different techniques to map organizational knowledge, and each technique can use a set of tools, approaches, objectives and specific characteristics (Jafari *et al*., 2009). In the following paragraphs, we show the main works that served as the theoretical basis for our mapping proposal.

Hansen and Krautz (2004) proposed using Rich Pictures (mechanism that uses pictograms for representation) as a technique to map the flow of organizational knowledge. The methodology consists of two large main stages: preparation phase and mapping phase.

- Preparation Phase: Based on the collected data, (s)he created an initial map of the organization.
- Mapping phase: It results in a knowledge map that describes actors and knowledge flow, as well as key features of the organization.

Hwang and Kim (2003) defined that a map is composed of two main components: diagrams that are graphical representations of components; and specifications, which are descriptions of the components. The authors also suggested creating a profile of the extracted knowledge, establishing a structure representing the characteristics of the mapped knowledge.

According to Kim and Hwang (2003), knowledge maps should achieve:

1. Formalization of all the knowledge inventories in the organization;
2. Perception of the relationship between knowledge;
3. Efficient Navigation of knowledge inventories;
4. Promotion of socialization/outsourcing of knowledge by connecting the experts' domains with knowledge explorers.

Eppler (2001) has developed five steps that must be performed to design and build a Knowledge Map. These are:

1st. Step: To identify the knowledge-intensive processes, problems or issues within the organization. The resulting map should focus on improving the intensive knowledge.

2nd. Step: To deduce the sources of knowledge, assets or relevant process elements or problems.

3rd. Step: To codify these elements in a way that it makes them more accessible to the organization.

4th. Step: To integrate this codified knowledge or documents information in a visual interface that allows the user to navigate or search for it.

5th. Step: To provide means for updating the Knowledge Map. A Knowledge Map is as good as the links it provides. If these links are outdated or obsolete, the map is useless.

The mapping techniques found in the literature show some approaches focusing on the flow of knowledge within the organization and the definition of knowledge sources. However, improved techniques may be applied to represent participants' knowledge based on knowledge flow.

Finally, Elias *et al.* (2007) proposed a methodology to identify and analyze knowledge flows in work processes. Such stages are:

1. To identify the main documents and people involved in the process;
2. To analyze the knowledge sources identified in the first step;
3. To identify how the knowledge and sources are involved in the activities performed in the process;
4. To analyze to find the problems that could be affecting knowledge flows identified.

The purpose of this paper is to integrate and improve these previous methods and generate a set of profiles of the participants in a software project team. From the data of these profiles, we can verify what is the most used knowledge by participants.

3 PROCESS OF KNOWLEDGE MAPPING IN SOFTWARE TEAMS

Our Process of Knowledge Mapping is mainly based on the work of Hansen and Kautz (2004), since their method allows enhancements or modifications. Furthermore, the work by Kim and Hwang (2003) contributes to the profiling strategy and the work by Eppler (2001) contributes to the definition of the steps to build the knowledge maps.

The main objective of the map is to find the core competencies of the participants based on their interaction with other team members and with sources of knowledge. The procedure of the Knowledge mapping consists of two phases:

- **Data Collection Phase**: The data that will compose the Knowledge Map will be collected. The collected data can come from two sources in the organization: the project or organization. Regardless of the origin, this phase will organize the data that will be employed to build a map of the structure;
- **Mapping Phase:** It is the organization of the data and the construction of the Knowledge Map. According to Table 1, the produced map is classified as a Knowledge Source Map, showing the sources of explicit (websites, books or documents) and tacit (participants) knowledge. Moreover, a profile for each participant will be produced, indicating his/her main accessed knowledge.

The moderator of the Knowledge Mapping Process can play many roles such as facilitator (during the data collection phase) or map developer (during the mapping stage).

3.1 Data Collection Phase

The purpose of the data collection phase is to extract the necessary information to create the Knowledge Map, as shown in Figure 1.

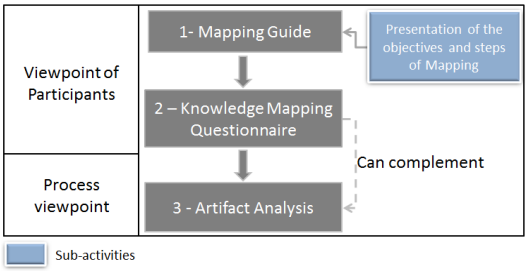


Figure 1: Activities of the data collection stage from the Knowledge Mapping process.

1. The Mapping Guide is a presentation showing the participants which activities they will do during the meeting. The purpose of the presentation is to support the facilitator of the meeting and present a practical visual guide to participants;
2. We apply the questionnaire to the participants who will create the Knowledge Map;
3. Analyzing Artifacts. The purpose of this activity is to see how organizations or group view the participants and to triangulate the facts with the questionnaire information.

We describe these activities in the following subsections.

3.1.1 Presentation of the Mapping Guide

The Mapping Guide should be presented to the participants of the meeting before the questionnaire. The structure of the presentation follows the following steps:

- Presentation of the Facilitator and his role for the group;
- Explanation of what is tacit and explicit knowledge;
- Brief explanation of Knowledge Management (optional);
- Brief explanation of what is Knowledge Mapping (if this is the first mapping);
- Presentation of the questionnaire structure;
- Presentation of the activity guides to the participants;
- Presentation of the questions on Knowledge Mapping.

Knowing the question of the knowledge mapping helps us to focus on the knowledge that we want and to capture accurate information, aiming to avoid extracting information that has nothing to do with the knowledge we demand.

3.1.2 Knowledge Mapping Questionnaire

The Knowledge Mapping questionnaire has a logical structure that seeks to find three aspects: what activities the participant exerted during the execution of the project, what or who (s)he researched to acquire knowledge and who (s)he helped.

Participants must be left free to consult each other, and they must have available resources to consult when they have questions while filling out the questionnaire. The reason for using these resources is that some people may not be able to remember some relevant information.

The first part of the Knowledge Mapping survey (see Figure 2) is related to the **Applied Topic of knowledge** of the activities (s)he carried out. The purpose of this information is to know what knowledge (s)he applied.

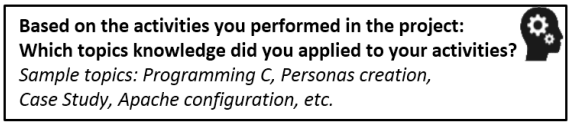


Figure 2: Field to describe which activities were conducted.

The field in Figure 3 is related to **Who /What (s)he consulted** to carry out his/her activities. The participant may indicate if (s)he consulted a person or an artifact and they should describe the name of the consulted person or artifact in the "Name of Person or Artifact" field. Then s/he must complement with a brief description regarding what was consulted. Some fields present different sizes because it might be possible that the participant has more than one consult to a device or person.

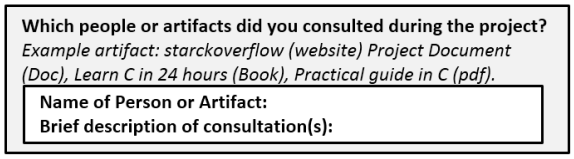


Figure 3: Field to describe the consults that were performed.

Finally, the participant must inform in the field shown in Figure 4 **Which people (s)he helped** during his/her activities. Based on this and the previous field, we can triangulate the information aiming to find the flow of knowledge among participants and to know what kind of knowledge takes place among them.

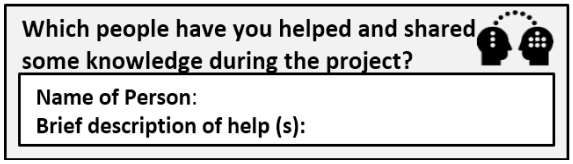


Figure 4: Field where the participant informs who (s)he helped.

3.1.3 Artifact Analysis

The artifact analysis is defined as the analysis of information from project-related documents that may be potential sources of knowledge. Its purpose is to explicit knowledge sources that will integrate the knowledge map.

3.2 Mapping Phase

The mapping phase will analyze the collected data in the data collection phase and will generate the knowledge map of the project team. Initially, we organize all the collected data on a table, as shown in Figure 5. Then, we produce the representation of the knowledge map sources (either by using physical materials with a whiteboard or through digital tools). Finally, we will generate the profile of each participant.

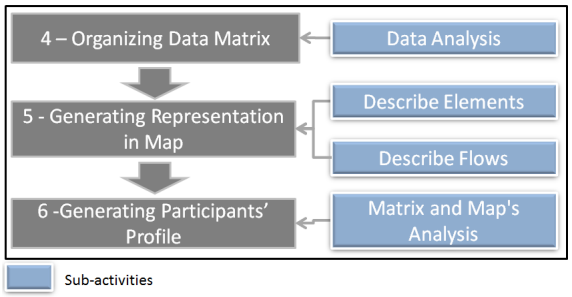


Figure 5: Activities from the Knowledge Mapping Phase.

3.2.1 Organizing the Data Matrix

Mapping questionnaires are analyzed at this stage and the moderator, who is implementing the Knowledge Mapping process, should examine each of them as (s)he carries out the parallel activities of this phase.

In the actors-artifacts relationship (where the actors are the participants), we organize all the data in a table following the format in Table 2. In the horizontal lines, we insert all the names of the project participants that have been mentioned in the fields "who you consulted" and "who you helped" from the questionnaires.

Table 2: Structure of Actors-Artifact in the Data Matrix.

Actors	Participant 1	Participant N	Artifacts	Artifact 1 (Type)
Participant 1		Id 1		Id 2
Participant N				

The columns are filled with the same name of the participants defined horizontally. After dividing the "artifacts", we can enter the names of the mentioned artifacts by any participant within the questionnaires.

While reading what artifacts were mentioned by the participants in the questionnaire, we should avoid duplication and then generalize when two participants refer to the same artifact. For example, two participants can mention the Stackoverflow online forum of questions and answers differently,

where one says "**Search Stackoverflow forum**" while another says "**stackoverflow.com**". Both participants refer to the same artifact, the Stackoverflow forum, so we will not insert two different columns for it. Instead, we can name the same column as "**Stackoverflow (website)**", where the parentheses in keyword help identifying what this artifact is.

After finishing to fill out the table, the cells are filled with an identifier of the description of the consulted information by the participant. For this, we will use a table for supporting where we will store the consult description gathered in the fields "who you consulted" and "who you helped" from the questionnaires. It is exemplified in Table 3.

Table 3: Structure to assist description of relationships.

Id	Relationship description	Participant
1	Description of Id 1	Participant 1
2	Description of Id 2	Participant 2

Finally, we have the name of the participants horizontally, while what they accessed (whether it is other participants or artifacts) is shown vertically.

3.2.2 Generating Representations in Map

The representation on the map can be done by a support tool which must have the following characteristics:

- Change colors or pictures of the node;
- Create edges between nodes;
- Assign weights and Text on the edges;
- Assign texts to the nodes.

After choosing the tool to be used, the activities of the process of knowledge mapping creation are initiated.

Based on the Data Matrix information built in the previous activity, we will perform the following steps to build the map:

1. Write what project members are;
2. Write what the artifacts informed by the participant are;
3. Center map members and leave the artifacts at the edges;
4. Insert an edge between nodes, namely between a member and an artifact, or between members;
5. Assign which or what are the relationships from such edge, based on the auxiliary table of the Data Matrix called Relationship Description;
6. Repeat from step 4 until all edges are created;
7. Document the map and its version.

After that, it is estimated that this map shows which members consult others and about what, and what artifacts are found during a project. It is recommended the review of the map by a second person in order to avoid omissions or errors.

3.2.3 Generating Participants' Profile

The profile of the participants is a representation indicating what skills or competencies (s)he is applying. They reference not only what (s)he informs, but what other participants inform. The map should also show how we can find him/her, what knowledge (s)he masters, what his/her sources of knowledge are and with whom (s)he communicates.

To generate the participant's profile, we will use the Data Matrix information, the analysis of the artifacts and the Knowledge Map as basis looking for:

- What are the main topics of knowledge (s)he employed in his/her activities?;
- What sources of knowledge does (s)he use?;
- What people has (s)he worked with or had some knowledge flow?.

This information will fill the items about the participant's profile in Table 4.

Table 4: Participant Profile structure.

Participant Name	<The full name entered by the participant.>
Position or Role	<Position or role of participant.>
Email	<Participant contact E-mail.>
Telephone	<Participant contact number.>
Keywords of major skills	
<Keywords that describe he/she skills. The keywords are the codes identified below.>	
Knowledge sources	
<What sources of explicit knowledge he/she consult based on the knowledge map.>	
People whom (s)he is related in the map	
<Which people the participant has a knowledge transfer based on knowledge map.>	
Worked projects within the Group	
<Project works within the research group.>	
Knowledge flow	
<The topics of knowledge informed by the participants.>	
Knowledge in…	
<Knowledge flow code.>	<Full description of flow.>

The fields Name, **E-mail**, **Position or Role**, and **Telephone** are extracted from the information previously collected. The information from the Knowledge Sources field will be collected analyzing

the data matrix based on the columns of the artifacts that the user entered. As shown in Figure 6, we use the participant's line and check the column of the artifacts used by him/her. This will be the information that will compose the field.

The **people to whom (s)he is related in the map** field will consist of all participants and people outside the project with whom the participant had any knowledge flow. In addition to identifying the participants, we assign weights according to the total sum of the flows between two participants, as seen in Figure 7.

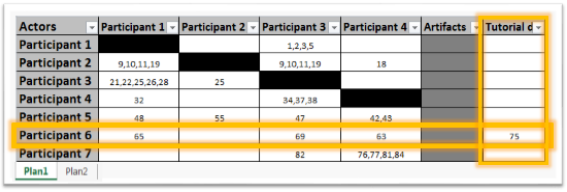


Figure 6: Capturing information about artifacts used by a participant.

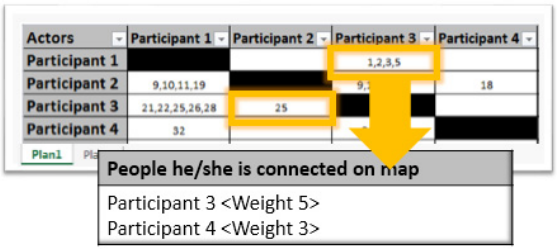


Figure 7: How to identify people connected to a participant.

Regarding the **Worked projects within the Group** field, this information will be extracted based on the analysis of the artifacts. In case that there is no identification, the field is filled with "None identified".

The **Participant Knowledge Topics** is the information that participants provided in the knowledge topic field of carried out activities in the questionnaire research, Subsection 3.1.2. After entering the information, we will generate codes for what was inserted. In addition, two descriptions may belong to the same code and thus increase the weight of this information, as seen in Table 5.

Knowledge flow will be the cross analysis of the Data Matrix for each participant (see Figure 8). The reason is that while the row shows just what the participant said, the column complements what others have reported about him/her. The Id (identifier) and his name should be placed in sequence in the field to be codified in the future.

Table 5: Knowledge topics of a participant.

Participant Knowledge Topics	
Review of material on Molic interaction modeling; Mockups together with Molic (diagrams); Case studies; Defects inspection; Inspection techniques for Molic diagrams; TAM (Technological Acceptance Model).	
Molic (3)	• Review of material on Molic interaction modeling; • Mockups together with Molic (diagrams); • Inspection techniques for Molic diagrams.
TAM (1)	TAM (Technological Acceptance Model.
Case studies (1)	Case studies.
Defects inspection (1)	Defects inspection.

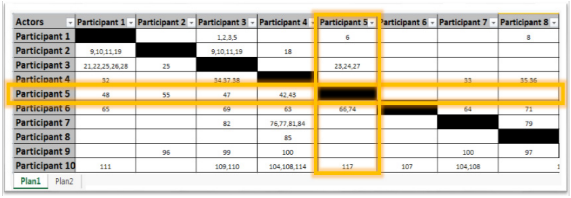


Figure 8: Way to capture the flow of knowledge from one participant.

After entering all the flows belonging to the participant, we will code with words that identify a concept or represent these flows (see Figure 9). The **Knowledge in …** field will be composed of all the coding of flows. Some encodings may have more than one flow, and the flow may belong to more than one coding.

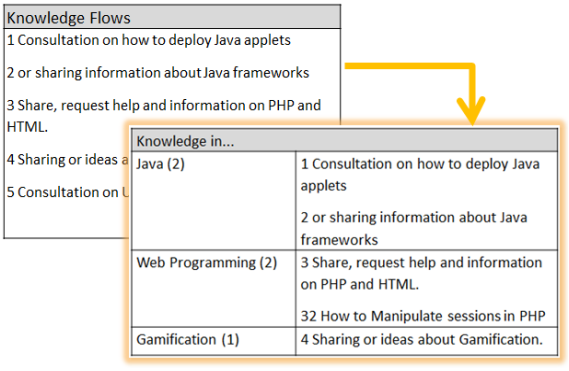


Figure 9: Analysis and codification of knowledge flows.

It is recommended the execution of codification by someone with knowledge of the organizational culture. Thus, the creation of codes is closer to

reflect the reality of the organization.

4 PILOT STUDY IN A RESEARCH AND DEVELOPMENT GROUP

The focus of the pilot study is to conduct a feasibility study of the Knowledge Mapping process. The primary purpose of a feasibility study is not to find a definitive answer, but to create a body of knowledge about the application of the technology (Mafra *et al.*, 2006).

As a result, we gain knowledge regarding if the process we are developing is feasible, if it produces a consistent result while identifying its limitations which, according to Shull *et al.* (2004), allows:

- The refinement of technology;
- The generation of new hypotheses on the application (in this case, the process of Knowledge Mapping) to be investigated in future studies.

The pilot study was applied in a software engineering and usability research group, which is formed by six Ph.D. candidates and four master students working on research and development in the areas of Software Engineering and Human-Computer Interaction. Thus, there are representatives of the population and, because it is a pilot study, we sought first to carry out the study within the research group and then evaluate in an industrial environment. The focus of the knowledge map was to find information related to types of knowledge that participants had applied or were applying in their research or in R & D (Research and Development) projects.

4.1 The Steps of the Pilot Study

The pilot study followed three steps detailed below.

1. Preparation: Contains the pilot study design, the creation of instruments and training of possible applicators of activity of Data Collection;
2. Implementation: The group in which the proposed technology would be applied attends a meeting in order to collect data. In this case, the Knowledge Mapping process;
3. Analysis and generation of results: The collected information will go through the data analysis of the Knowledge Mapping process.

By running the pilot study, we can verify the main aspects required for the application of the proposed technology (the process of mapping of knowledge) and analyze its limitations to evolve it in the future.

4.2 Preparation

In this phase, we plan and prepare all the instrumentation and contact the people that are necessary for the implementation of the Knowledge Mapping process. The main purpose of the preparation is to address threats to validity. Based on the recommendations by Wohlin *et al*. (2012), the following threats were addressed:

Internal Validity (Instrumentation): This is the effect caused by the artifacts used in the execution of the experiment. In the case of a poorly-planned experiment, its results will be negatively affected. Thus, a second researcher reviewed the artifacts created by the author process.

Construct Validity (Expected Experimenter): The author of the knowledge mapping process can consciously or unconsciously cause bias in the results of a study based on what (s)he expects the results of the experiment will be. When implementing the experiment, we asked another researcher with no involvement in this research to apply the process. However, in the analysis phase and the generation of results, the author of the process performed the analysis.

External Validity (Interaction of Participants and Treatment): It occurs when a sample does not represent the population we want to generalize. The focus of the process is to map software project teams. We chose a research group and R & D (Research and Development) projects due to convenience and the similarity of their themes and situations.

4.2.1 Instrumentation

For the pilot study, the following instruments that supported the whole process were developed:

Approach Manual: a Knowledge Mapping process manual was prepared explaining step by step how to apply and generate a knowledge map, how to collect data, which tools to use and what the end products of the process would be.

Knowledge Mapping Questionnaire: a questionnaire that aims to capture key information needed to generate the knowledge map and profiles of the participants.

Presentation of the Mapping Guide: a presentation guide that supports the moderator when applying the questionnaire and participants during the data collection. The presentation consists of 12

slides that show the objectives of the data collection, the structure of the questionnaire and a behavior guide for participants to follow during the session.

4.2.2 Guest Researcher

A researcher with no relation to the research was asked to administer the questionnaire to the participants. At a meeting, the author of the proposal presented the research objectives, the guide of the approach and the tools (questionnaire and presentation) for the guest researcher.

Additionally, we collected suggestions from the invited researcher to better conduct the experiment, which allowed gathering initial feedback for the improvement of the technical instrumentation. After the transfer of information, the execution of the study was scheduled with the group of participants.

4.3 Execution

Execution is the application of knowledge mapping questionnaire with the participants that will create the knowledge map. The questionnaire was printed and distributed to participants with no time limit to fill it out, and we allowed the interaction among them. The guest researcher assumed the role of facilitator, which sought to conduct all data collection and answer questions from the participants.

The participants took around thirty minutes to answer the questionnaire. The author of the proposal was absent during the execution process of the data collection in order to avoid any bias in the pilot study. After finishing the execution, the data was delivered to the author of the process for analysis.

4.4 Analysis and Generating Results

We explain the performed data analysis in this section. The results are related to the knowledge map of the team and the profiles of participants. For the execution of this phase, we did not invite another researcher, because the process needed a closer analysis from the authors of the proposal.

At this stage, all the Knowledge Mapping phase must have been executed, as described in Subsection 3.2, for the activities of **Organizing Data Matrix** and **Generating Representation in Map**.

For the **Generating Participant´s Profile** activity, which is the analysis and creation of all profiles, there is no reliable estimate to be informed due to the improvement of the technique while performing the activity. We explain the results of this pilot study in the following section.

5 RESULTS

This section presents the results of the implementation of the Knowledge Mapping process. In addition, lessons learned and results of the implementation of the knowledge mapping process are presented.

5.1 Knowledge Mapping Results

As presented in Section 4.3, in the execution of the study, we employed a printed questionnaire (subsection 3.1.2) with ten participants in an R & D (Research and Development) group. Ten questionnaires were analyzed in the mapping stage. A spreadsheet was used to support the creation of the Data Matrix.

For the matrix, two tabs have been created. The first one shows the connections between participants with participants or artifacts, as described in Subsection 3.2.1. A sample result can be seen in Figure 10.

Actors	Participant 1	Participant 2	Participant 3	Participant 4	Participant 5
Participant 1			1,2,3,5		6
Participant 2	9,10,11,19		9,10,11,19	18	
Participant 3	21,22,25,26,28	25			23,24,27
Participant 4	32		34,37,38		
Participant 5	48	55	47	42,43	
Participant 6	65		69	63	66,74
Participant 7			82	76,77,81,84	
Participant 8				85	

Plan1 Plan2

Figure 10: First tab of the Data Matrix.

The second tab stores the description Ids generated in each cell. Moreover, it stores the participant's name and if the data is going in or out Figure 11).

Id	Description	Helped(<)/ Consulted (>)	Participant
1	Help in the division or material for interaction modeling	<	Participant 1
2	Sharing information on technical proposals for inspection Mo	<	Participant 2
3	Sharing and request help information about GT and TAM.	<	Participant 3
4	Sharing ideas about gamification.	<	Participant 4
5	Consultation on TAM's information and GT	>	Participant 5
6	Consultation on statistical test information	>	Participant 6

Plan1 Plan2

Figure 11: Second tab of the Data Matrix.

Then, we generated the graphical representation of the Knowledge Map based on the steps described in Subsection 3.2.2. We applied the NetMiner 4.2.1 tool due to its ease of use. The generated result can be seen in Figure 12.

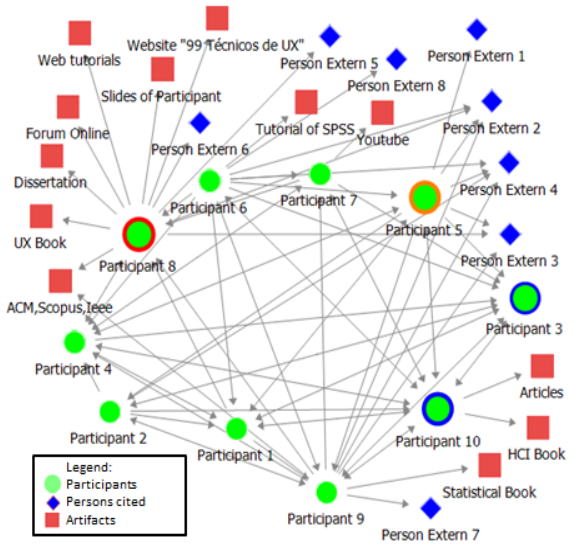


Figure 12: Group map generated by NetMiner (available at: http://www.netminer.com/).

The map elements were created based on the Data Matrix. As recommended in the approach's manual, participants were centralized on the map and indicated people or artifacts were allocated at the edges of the map.

Knowledge maps can provide a set of knowledge sources and flows. In addition, managers can use this information for decision-making. However, it is important to carry out a systematic analysis of such knowledge maps to reveal relevant insights of the organization (Chan and Liebowitz, 2005). Consequently, we applied Social Network Analysis (SNA) to systematically investigate some aspects of knowledge flow depicted by the knowledge maps.

In the map, we identified two central connectors. The central connectors are people with whom other participants interact more (Cross and Prusak, 2002), they are the participants from 3 to 10 (green circle with a blue border in Figure 14). Participant 5 is classified as Border Key (Cross and Prusak, 2002), which communicates with more people outside the network and serves as an ambassador between the network's internal and external knowledge.

We can check the level of reciprocity that is the similarity between the entries of two participants (Tichy *et al.*, 1979). The strongest connections are between the participants 1 and 3, followed by the participants 9 and 10.

Additionally, we can analyze that Participant 8 behaves like a person with the most access to artifacts (Red border in Figure 14). Moreover, Participant 5 (Orange border in Figure 14) is the person who consults the higher number of people within the network, which may be an indicator that (s)he had the current highest level of learning.

After creating the knowledge map graphically and in the matrix, we analyzed and generated the profiles of the participants based on the steps of Subsection 3.2.3. We define the key words that represent the main competences of each participant and using such information, we identified his/her and the group's main knowledge. Table 6 presents a profile created for one of the participants.

Table 6: Profile from a participant.

Participant Name	Participant 10
Position or Role	PhD student
Email	XXX
Telephone	XXX
Keywords of major skills	
Systematic Literature Review (6), Paper Writing (4), Statistical Analysis (3) Usability (3) Pilot Study (3), Modeling themes (3), Review proposal (3).	
Knowledge sources	
• ACM; • Scopus; • Ieee; • Books of HCI.	
People whom (s)he is related in the map	
• Participant 9 < Weight 7> • Participant 6 < Weight 7> • Participant 4 < Weight 4> • Participant 3 < Weight 4> • Participant 3 < Weight 3> • Participant 5 < Weight 3> • Participant 2 < Weight 2> • Participant 1 < Weight 2>	
Worked projects within the Group	
None Identified	

Finally, we produced the two main products of the Knowledge Mapping process: the group's knowledge map and a set of profiles for each participant.

Group leaders received the data for analysis and assessment. Moreover, the analysis of the participants, based on the maps and in the matrix, includes: who accessed other participants, who accessed more artifacts, which participants had the strongest connection (edges or knowledge flow) and what the strongest knowledge domain of the group was.

5.2 Lessons Learned of the Knowledge Mapping Process

We requested the participants to answer the questionnaires based on the main question of the mapping. Thus, the questionnaire words and

examples should be according to the defined mapping question.

The participants must be free to communicate with each other, so that they can easily retrieve information when filling out the questionnaires. Research on books, websites or document names should be allowed for a richer filling of the questionnaire.

During the mapping step, the matrix was modified based on the original idea with respect to the field describing the relations. A column with the name of the participants was inserted to provide a better way of identifying who owned that description in a bigger data set.

Once completing the knowledge map, we started the creation of the profiles from the participants. In the beginning, the first version of the proposed structure did not work to generate the profile of the participants. This was due to the lack of a review process of the results for filling fields correctly.

Improvements in the participants' profile form were: 1) The structure has been redesigned to display necessary information from each participant profile. 2) A knowledge technique for identifying the applied knowledge of the participants was defined to analyze the flow of knowledge among the participants. 3) The steps of the analysis and profile creation activities have been rewritten. The main goal for such change is that others can properly apply the process without help or interference of the authors of the process.

6 CONCLUSIONS AND FUTURE WORK

The Knowledge Mapping process presented in this paper maps a group of participants and creates profiles for each participant. In addition, we carried out a pilot study where it was found that this process is feasible.

Each profile displays, besides basic information on how to find the participant in the organization, with whom (s)he is connected to on the map and what activities (s)he performs. The profile also displays indicators of the main competences (s)he is carrying out in the group using information that other participants employ from him/her.

The executed knowledge mapping process within the study produced a map where one can check which connections a participant has with each knowledge source, either being explicit (websites, books, and so on) or tacit (access to people). Also, it is possible to check on each edge which knowledge is flowing.

The advantages found to justify the creation of a knowledge map in the study are:

- To check what main competences a participant is in fact executing. Based on this, we can verify if (s)he is applying something for which (s)he was designed or if there are any mistakes in the execution of his/her activities;
- To check for anomalies in the knowledge flow of a participant. Perhaps a participant is requiring a source of knowledge that does not fit into his/her roles. It can mean a learning signal or irregularity;
- To check if the flow of information between members is happening. In an integrated team, we can see through a map if two members are or not interacting when they should be. For example, the analyst responsible for gathering requirement and the developer;
- To identify the current knowledge in a group or software team. Based on the identified keywords within the profiles, we can draw conclusions from what knowledge the group or team is employing and which have high scores.

Finally, as future work, we intend to:

- Apply the Knowledge Mapping Process in a Case Study with software projects teams;
- Automate the data analysis process and the creation of profiles;
- Compare Knowledge Mapping with network analysis techniques such as Social Network Analysis;
- Apply the Knowledge Mapping Process in a Knowledge Audit Process as Elias *et al*. (2010).

ACKNOWLEDGEMENTS

We would like to thank the support granted by CAPES; by FAPEAM through processes: 062.00600/2014; 062.00578/2014 and by CAPES process AEX 10932/14-3. We would like to thank all the subjects who participated in this study.

REFERENCES

Alavi, M., and Leidner, D. E., 2001. Review: Knowledge management and knowledge management systems:

Conceptual foundations and research issues. In *MIS Quarterly,* v. 25, n. 1, p. 107-136.

Anandarajan, I., and Akhilesh, A. K., 2012. An exploratory analysis of effective indo-Korean collaboration with intervention of knowledge mapping. In *Proceedings of the 4th international conference on Intercultural Collaboration*, p. 129-132. ACM.

Balaid, A.S.S., Zibarzani, M., and Rozan, M.Z.A., 2013. A comprehensive review of knowledge mapping techniques. In *Journal of Information Systems Research and Innovation (JISRI)* v. 3, (p. 71-76).

Bhatt, G. D., 2001. Knowledge management in organizations: examining the interaction between technologies, techniques, and people. In *Journal of knowledge management*, v. 5, n. 1, p. 68-75.

Bjørnson, F. O., and Dingsøyr, T., 2008. Knowledge management in software engineering: A systematic review of studied concepts, findings and research methods used. In *Information and Software Technology*, vol. 5, n 11, p. 1055-1068.

Chan, Kelvin, and Jay Liebowitz. 2005. The synergy of social network analysis and knowledge mapping: a case study. International journal of management and decision making 7.1, p. 19-35.

Cross, R., and Prusak, L., 2002. The people who make organizations go-or stop. Harvard business review, vol. 80, n. 6, p. 104-112.

Davenport, T. H., and Prusak, L., 1998. *The Book.* Working Knowledge: How Organizations Manage What They Know, Harvard Business School Press, Boston, MA, USA.

Davenport, T.H., and Völpel, S.C., 2001. The rise of knowledge towards attention management. In *Journal of Knowledge Management*, vol. 5, n. 3, p. 212 - 222.

Eppler, M. J., 2001. Making knowledge visible through intranet knowledge maps: concepts, elements, cases. In *System Sciences*, Proceedings of the 34th Annual Hawaii International Conference. IEEE, p. 189-205.

Elias, O.M.R., García, A. M., Vara, J. F., Vizcaíno, A., and Soto, J. P. (2007). Knowledge flow analysis to identify knowledge needs for the design of knowledge management systems and strategies: a methodological approach. In *Proceedings ICEIS 2007-9th International Conference on Enterprise Information Systems*, p. 492-497.

Elias, O.M.R., Rose-Gómez, C.E., Vizcaíno, A., and Martienz-Garcia, A. I. (2010). Integrating current practices and information systems in KM initiatives: A knowledge management audit approach. In *Proceedings of the International Conference on Knowledge Management and Information Sharing (KMIS),* p. 71-80.

Gourova, E., Toteva, K., and Todorova, Y., 2012. Audit of Knowledge flows and Critical business processes. In *Proceedings of the 17th European Conference on Pattern Languages of Programs*, ACM, p. 10.

Hansen, B. H., and Kautz, K., 2004. Knowledge mapping: a technique for identifying knowledge flows in software organisations, Springer Berlin Heidelberg, (p. 126-137).

Jafari, M., Akhavan, P., Bourouni, A., and Roozbeh, H. A., 2009. A Framework for the selection of knowledge mapping techniques. *In Journal of Knowledge Management Practice*, v. 10, n. 1, p. 9.

Krbálek, P., and Vacek, M., 2011. Collaborative knowledge mapping. In *Proceedings of the 11th International Conference on Knowledge Management and Knowledge Technologies*, ACM, p. 29.

Kim, S., Suh, E., and Hwang, H., 2003. Building the knowledge map: an industrial case study. In *Journal of knowledge management*, 7(2), p. 34-45.

Lindvall, M., and Rus, I., 2002. Knowledge management in software engineering. In *IEEE software, v. 19*, n. 3, p. 26-38.

Lindvall ,M., Rus ,I., and Sinha, S.S., 2003. Software systems support for knowledge management. In *Journal of Knowledge Management,* vol. 7 Iss: 5, pp.137 - 150.

Mafra, S. N., Barcelos, R. F., and Travassos, G. H., 2006. Aplicando uma metodologia baseada em evidência na definição de novas tecnologias de software. In *Proceedings of the 20th Brazilian Symposium on Software Engineering (SBES 2006)*, vol. 1, p. 239-254). *(In Portuguese).*

Mitchell, S. M., and Seaman, C. B., 2011. A knowledge mapping technique for project-level knowledge flow analysis. In *Empirical Software Engineering and Measurement (ESEM)*, International Symposium on, IEEE, p. 347-350.

Nada, N., Kholief, M., and Metwally, N., 2009. Mobile knowledge visual e-learning toolkit. In *Proceedings of the 7th International Conference on Advances in Mobile Computing and Multimedia*, ACM, p. 336-340.

Nonaka, I., and Takeuchi, H., 1995. *The Book*. The Knowledge-Creating Company, Oxford University Press, New York.

Tichy, N. M., Tushman, M. L., and Fombrun, C., 1979. Social network analysis for organizations. Academy of management review, vol. 4, n. 4, p. 507-519.

Vasconcelos, J.B., Seixas, P.C., Lemos, P.G., and Kimble, C., 2005. Knowledge Management in Non-Governmental Organisations: A Partnership for the Future. In *Proceedings of the 7th International Conference, Enterprise Information Systems* (ICEIS), Miami, USA, p. 1-10.

Shull, F., Carver, J., and Travassos, G. H., 2001. *The Book*. An empirical methodology for introducing software processes. In *ACM SIGSOFT Software Engineering Notes*, vol. 26, n. 5, p. 288-296.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., and Wesslén, A., 2012. *The Book.* Experimentation in software engineering. Springer Science & Business Media.

Agile-similar Approach in Traditional Project Management
A Generalisation of the Crashing Model

Dorota Kuchta[1], Pierrick L`Ebraly[2] and Ewa Ptaszyńska[1]
[1]*University of Technology, Wrocław, Poland*
[2]*Ecole Nationale des Ponts et Chaussees, Paris, France*
{dorota.kuchta, ewa.ptaszynska}@pwr.edu.pl, plebraly@gmail.com

Keywords: Time-cost Trade-off, Project Crashing, Linear Programming, Agile Development, Decision-making Tools.

Abstract: In numerous industries such as software development there has been increasing pressure on the supplier to provide early results. In this study we propose a method to adapt traditional project scheduling in order to meet early expectations of the client while limiting costs. First we present the philosophy of the agile methodologies in which meetings with stakeholders play an important role. Therefore it is valuable to take them into account in order to develop new models. Based on it we present a proposal of Linear Programing (LP) model which goal is to minimize the crashing cost and maximize customer satisfaction. In our model we distinguish activities that are rewarded (can increase customer satisfaction) if they are completed before certain meetings. What is more, we assume that project's budget can be modified during meetings. At the end we present an example of using the proposed model.

1 INTRODUCTION

Recently we have been dealing with two types of project management approaches (Wysocki, 2014) - traditional approaches and agile approaches. Simplifying, it can be said that the main difference between them is that in the traditional approach the project scope, deadline and budget are essentially known at the beginning and the project management methods aim at realizing this scope while in the agile approach, the scope, and even the goal tend to change during the project realisation and the project management methods are aimed at helping the project team to adapt the project and its realisation to the changing objective. However, the two approaches cannot be isolated from each other. Recently, researchers have noticed the need to compare and combine the two approaches.

(Kosztyn, 2015) proposes a matrix-based approach to project planning and describes a generic algorithm that builds schedules for both agile and traditional project management approaches. (Spundak, 2014) compares both approaches and suggests that a mixed approach may be needed in the future as we have been facing a more and more varied spectrum of project types and, to use his words, methodology should be adapted to the project and not vice versa. This paper continues this line of research, as it allows introduction of agile elements into traditional project management. In Figure 1 and Figure 2 both approaches (traditional and agile) are described; the upper part of triangle represents objectives while the lower parts represent the chosen set of constraints. These are widely inspired by a similar representation found in (Kosztyn, 2015).

Figure 1 presents agile approach. A simple way to understand agile approach is to see it in terms of maximizing goals in a fixed time and cost environment.

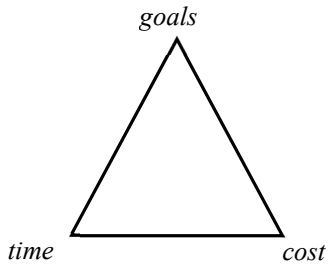


Figure 1: Short term approach, agile approach.

Figure 2 presents traditional approach. Traditional planning focuses on reaching fixed goals while trying to minimise time and cost; which implies solving a time-cost trade-off problem.

Kuchta, D., L'Ebraly, P. and Ptaszyńska, E.
Agile-similar Approach in Traditional Project Management - A Generalisation of the Crashing Model.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 318-326
ISBN: 978-989-758-187-8

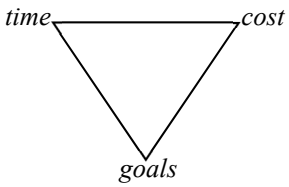


Figure 2: Long term approach, traditional approach.

In this paper we will present an approach to introduce an agile element into traditional approach.

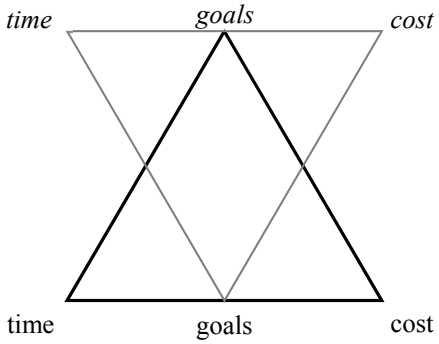


Figure 3: Mixing both approach.

The approach we propose here allows modification in the project short-term objectives in order to deal with another major aspect of project planning - client satisfaction.

Our proposal is motivated by the fact that the customer plays a growing role in project management. In a competitive environment it is important to keep in mind that a major criterion for project success is customer satisfaction (Al-Tmeemy et al., 2010). This aspect is made clear in the agile approach to project management, where the customer is allowed to change his expectancies between every sprint as to the project expected outcome. Thus, the aim of our proposal is to allow the customer to influence the project realisation in the traditional approach to a greater degree than it is usually done. We blend in the traditional approach to project management the idea taken from agile management of regular meetings, where customers should be ”made as happy as possible” (Al-Taani et al., 2013).

In traditional project management minimizing cost or time, or more often achieving an optimal cost-time trade-off (Alba, 2007) is a common goal and has been tackled by different methods, including neural networks (Dohi et al., 1999) Linear programming models have been used for a long time to model the dependencies in the project network and their consequences for the project schedule. The crashing model is a well-known model in formal traditional project management, i.e. (Abbasi et al., 2011). Traditional project crashing is a method for shortening project duration by reducing the duration of project activities situated on the critical path. It allows, in the scheduling phase, to decide which project activities should be crashed shortened at a cost in order to achieve a desired project completion time while keeping the cost minimal (or to find the earliest possible completion time given a global budget for activities shortening, a problem that is outside of the scope of this article). The desired project completion is dictated by the customer.

We thus propose to complete the crashing model with a measure of customer satisfaction, implemented through regular meetings with the customer.

This paper proposes a LP model which should help deciding which activities should be crashed to ensure the client is satisfied throughout the project realisation, and not only after project completion, to the highest possible degree at the minimal cost in a budget and time limited environment.

The proposed set of constraints provide time and money limits trough-out the project evolution, as well as a reward system to encourage early satisfaction of client’s needs.

2 CLIENT INTERACTIONS IN DIFFERENT CONTEXTS

We made frequent use of meetings with client to define time-cost trade-off. The notion of clients and meetings can be given a more general definition.

Definition 1. *A client is the person or the organization who/which is the addressee of the project product, interested in the development of the project for future work.*

Note that there could be many stakeholders of the project. Here we concentrate on the principle stakeholder, as different stakeholders may have conflicting interests (Freeman, 1984). Focusing on handling the expectancies of different stakeholders could be an object of future research.

Definition 2. *A meeting is a point in time at which the client has access to the state of the project, and thus is able to measure the state of advancement of the project according to its own criteria. Having taken knowledge about the current project state, it is possible that after the meeting the client changes its priorities, induce more resources for the future of the project or even redefines the project.*

Meetings are important in that they are the tools that allow clients to have insight and impact on the

project. In the planning stage we try to model the customer ongoing satisfaction after each meeting according to the information possessed in this moment, but after each meeting the customer has the right to change its mind, which implies the model must be reapplied to the unaccomplished part of the project with changed parameters each time the client express a change in priorities.

Definition 3. *The project end date is the furthest possible date at which the project stops. Three reasons may cause a project to stop: project completion, when no further activities are to be completed, the end of the current planning horizon for the project, that is, a new planning phase needs to be planned later before this point so the project can carry on, or the decision to give up on the project because it has taken too long to give expected results.*

For some projects, end date considered as a hard deadline is quite unsatisfactory, as the project outcome is not clearly perceived at the beginning of the project. In this case, the project end date should be referred to as the planning horizon and the role of the finish activity is modified as project success does not require any given activity to be completed.

We will now discuss three project examples, showing how different the role of the client and form take by the meetings can be:

1. *A Road-construction Project*. In this type of project, the client is the sponsor for the road-work, has little interest for the project inner development, and has strong expectations for the project completion date. For this reasons, all meetings are set after the would-be finishing date of the project, at which point strong penalties will be handed for not completing the project in time. In this case, the project end date would represent the latest possibly conceivable time for finishing the project.
2. *A Not-for-profit Open-source Software Project.* In this radically different project, the aims of the team developing the software is to deliver a software that meets their own expectations, while trying to get other developers to join their development team. For this reason, the client is any hypothetical developer potentially interested in giving time to the project, including but not limited to those actually handing in code for the project. Costs should be expressed in terms of man-hours rather than any other unit, and bounties represent further involvement from developers in the next development run, either through current project developers deciding to give in more time for the project, or other developers joining the project. Meetings here are public releases for the software, which correspond to the time at which each developer can benefit from the work done on other parts of the project, while Tend is the date at which the geometry of the development team is anticipated to change.
3. *A Commercial Software Project for an External Customer.* Here, the customer pays for the development of the product, but he may stay very imprecise during project planning and even during the early stages of project realisation. Though he is often unaware of this, the customer is not sure about what he wants the product to address. However, there are usually certain functionalities and concerns that the customer wants with certainty, and that would keep him satisfied during early project development. If the team can make his early satisfaction high enough, then the customer will be more inclined to accept future failures or constraints while carrying on with the project. Thus, it is vital in this type of project to ensure early satisfaction is reached by presenting achievements straight away even though some important aspects of the final product are yet to be decided or even identified.

3 PROPOSED MODEL

3.1 Basic Definitions

A project P can be broken down in a number of activities that can be either tasks or events, an event being an activity of zero duration by opposition to a task which has to be performed (Elkadi, 2013). In the rest of this paper we will use the word activity. For the purpose of this model, we suppose that each task and event is performed at most once in the course of the project. Two additional dummy events are added to provide the beginning and the completion of the project. Let $V = [\![0, N+1]\!]$ be the set of activities (Freeman, 1984).

For each of these activities we define a base duration and cost, using notations that are consistent with those proposed. This is done by introducing vectors $\mathrm{D} \in \mathbb{N}^V$ and $C \in \mathbb{R}^V$, where D_i and C_i are respectively the duration and the cost of performing activity i. Additionally each activity can be crashed by devoting more resources to it. This increases the total cost to perform the activity. We introduce vectors $D^C \in \mathbb{N}^V$ and $C^C \in \mathbb{R}^V$ as, respectively, the maximal crashing duration (i.e. by how many days we can maximally shorten the activity) and the daily cost of crashing activities (the cost of shortening a given activity by one day).

Crashing cost is always positive, and we have $0 \leq D_i^C \leq D_i$ for each activity. If $D_i^C = D_i$ then we say the activity can be externalized.

Another concern that has to be addressed is the relationship between activities. For this purpose, we introduce a graph A in which each arc indicates that the predecessor activity must be completed before starting the successor activity. Unlike (Brucker, 1999) we will not introduce waiting times between two activities, as those can be modelled by adding additional dummy activities with a fixed duration between activities that need to be separated, which means that the graph can be seen as a subset of $V \times V$, at the cost of over-dimensioning the set V.

Actual starting times are kept in a variable vector $t \in \mathbb{N}^V$ and actual crashing times are kept in a variable vector $c \in \mathbb{N}^V$.

3.2 The Linear Programming Model proposed – A General Description

Three different objectives have to be taken in account in order to plan a project: reaching goals, as perceived by the client, the time needed to do so and the money used to do so. However, these are all dependent on each other and for this reason, as described above, past approaches made some of them constraints and other objectives. It is also important to take into account, as mentioned above, the objective of customer satisfaction which is absent in classical crashing models.

On the one hand, our model focuses on two resources: time (the project completion time), and money (budget available for crashing activities and carrying tasks), to reach a fixed goal. What is more, we keep track of project's achievement throughout the duration of the project by introducing meetings with the client to take into account client satisfaction at different points during the project, because we assume that the customer, though interested in the development of the project, does not need to be aware of every activity but rather has knowledge about the state of a project at a number of given times (meetings) during the project.

In order to address the three objectives: early completion time, minimal cost and maximal ongoing customer satisfaction, we decided:

- to make the time objective a constraint - a project deadline will be imposed;
- to make the cost objective both a model objective (the total cost of activities crashing should be minimal) and a constraint: in each consecutive period between two meetings with the customer there is a budget available for carrying out activities, including crashing;
- to make the customer satisfaction a model objective. For this reason, the objective function will focus on money.

Thus, our model will have two objectives: the total cost of crashing the activities (minimised) and the satisfaction of the customer throughout project realisation (maximised). The latter objective is difficult to express in a formal way and to measure, as it is immaterial. We have decided to measure it in monetary units - project planners will be asked to express the customer satisfaction in terms of value. This translation is crucial, as it will play an important role when we combine the objectives, which will be discussed later on. The problem of expressing consumer satisfaction in monetary units will be discussed further in the next subsection.

To account for time-wise gain in this objective function, bounties are awarded for early activity completion. These bounties are awarded for each activity if the given activity is started before the j-th meeting with the client. For this we introduce $W \subset \mathbb{N}$ the set of meetings and $E \in \mathbb{N}^W$ the vector of meeting dates. B_i^j is the bounty awarded for activity i if it has completed before meeting j. For the purpose of calculation, we use a matrix $B' \in \mathbb{R}^{V \times W}$ where $B'^j = B^j - B^{j+1}$. This comes in handy as we can now attribute a bounty for meeting j without checking whether or not we already gave a bounty for week $j-1$. However, it can be noted that activities that are not valued by the client, or activities that need to be completed in order to have a value for the client, must be treated with caution. In the first case no bounty should be awarded for the activity, and, in the second case, bounties should be awarded for an event that depends on and only on the completion of the activity. Things such as finishing a user interface, or giving a functional preview to the client, should be modelled through events and awarded with big bounties, as even if they have shallow meaning in terms of work-involvement, they play a great role in client satisfaction. It needs to be remembered that bounties are not used to congratulate a team on its fast work, but to represent the value-added of having the client implied in the development. For this reason, bounties should be calculated based on their capacity to get the client further involved in the project.

We obtain a bi-criteria linear programming problem, which can be solved in many ways. Here we assume the weighting approach with equal weights given to both objectives, but of course the approach to solving the bi-criteria problem could be changed, either by modifying the weights or by using a different method to combine the criteria.

Table 1: Notations previously introduced.

Name	Type	Notes
N	natural number	The number of activities to be performed throughout the project, including dummy activities
V	subset of $\mathbb{N}$	Project activities, including start and finish activities
A	subset of $V \times V$	Activity dependency graph. If (a,b) is present in the graph then b depends on a to start
W	subset of $\mathbb{N}$	Meetings
D	element in $\mathbb{N}^V$	Vector of the base durations in time unit for each activity
D^C	element in $\mathbb{N}^V$	Vector of the maximum crashing durations
C	element in $\mathbb{R}^V$	Vector of the base costs for each activity
C^C	element in $\mathbb{R}^V$	Vector of the crashing cost per time unit for each activity
MT	element in $\mathbb{N}^W$	Vector indicating the times on which meetings take place
$Tend$	natural number	Hard limit for project completion (note this can also denote the project planning horizon)
B (B')	element in $\mathbb{R}^{V \times W}$	Matrix of the bounties handed out for completing a given activity before a given meeting.
M (M')	element in $\mathbb{R}^W$	Vector used to represent budget limits in the span between two meetings
y	variable in $\mathbb{N}^V$	Calculated starting time
x	variable in $\mathbb{N}^V$	Calculated crash duration
o	variable in $\{0,1\}^{V \times W}$	Binary indicating whether an activity is started before a given meeting

As mentioned above, resources availability for activities crashing is limited in time which is modelled using a fixed budget limit for each interval between two meetings, M_j. We use a construction similar to the one used for B' to deduct a matrix M' which can then be used to account for staying in the budget during intervals $[0..j]$.

A variable has to be introduced to denote whether activity i has started before meeting j. This variable matrix is noted (o_j^i) and calculated dynamically.

3.3 The Linear Programming Model proposed – Mathematical Formulation

In this subsection we will present the mathematical formulation of the model described in the previous section. Below we present a table summing up all notations used to this point.

This leads us to introduce the following LP model:

$$\text{Min cost} : \sum_{i \epsilon V} \left(C_i^c \cdot x_i - \sum_{j \epsilon W} o_i^j \cdot B_i^j\right) \quad (1)$$

$$\forall j \in W, \sum_{i \epsilon V} o_j^i \cdot Ci \leq Mj \quad (2)$$

$$\forall i \in V, x_i \leq D_i^C \quad (3)$$

$$y_0 = 0 \text{ and } y_{n+1} \leq T_{end} \quad (4)$$

$$\forall (i,j) \in A, y_j - y_i + x_i \geq D_i \quad (5)$$

$$\forall_{i \epsilon V, j \epsilon W}\ 10000 \cdot o_j^i \geq MT_j - y_i \quad (6)$$

$$\forall_{i \epsilon V, j \epsilon W}\ 10000 \cdot (1 - o_j^i) \geq -(MT_j - y_i) \quad (7)$$

Equation (1) is the objective function and equations (2) - (7) represent constraints. Constraint (2) refers to budget limits in the timespan between two meetings. Equation (2) is there to make sure the project does not fail because of solvability limits. It is important only for project that have a big monetary impact on the firm. For those projects it is important to keep in mind that a project could be profitable but run through a phase in which is not solvable. Constraint (3) refers to maximum crashing durations of specific activities. Constraint (4) refers to starting and finishing time of the project. Constraint (5) refers to the order of activities. Constraints (6) and (7) are used for indicating whether an activity is started before a given meeting. The '10000' is a more or less arbitrary constant in order to be able to linearize 'if' tests. Note that in equation (1) two different objectives were accounted for in an equation. The sum here is used because matrix B can always be normalized to be of the same order of magnitude as the costs in the project, as it is used there and only there. However, the decision of whether or not to normalize the bounty matrix has to be taken when this matrix is filled: in some situations the benefits of showing early results to the client are not commensurable to the crashing costs involved, and, in these cases, no crashing should occur. On the other hand, sometimes costs are not an issue if the client can have early results, for example while handling a project designed at resuming production for a much

larger manufacturing scheme, only early delivery should be valued.

4 EXAMPLE

In this section we go through a few cases in which the introduced model results in a different schedule chosen for the project development than in a classical approach. Due to the limited space in this article we present only a brief example of a project. The proposed linear programming model has been implemented in a free editor GUSEK (GLPK Under SciTE Extended Kit) and tested on a selected research project performed by Wroclaw University of Technology. The main goal of the analysed project is to identify success and failure factors of research projects with particular emphasis on projects performed at universities, based on the example of Poland and France.

Identifying success and failure factors of projects is a popular problem in the scientific literature of project management, i.a. (Blumer et al., 2013), (Elkadi, 2013),, (Kosztyn, 2015), (Zou et al., 2014). Research projects at universities represent very specific type of project, i.a. (Luglio et al., 2010), (Powers et al., 2009) therefore they require dedicated research.

In the analysed project the following activities were identified:

1. Preparing IT tools to support project realization.
2. Collecting contacts among stakeholders of research projects in Poland.

Collecting contacts among stakeholders of research projects in France.

3. Studying literature.
4. Conducting a survey among stakeholders of research projects in Poland.
5. Performing interviews with specific stakeholders of research projects in Poland.
6. Conducting a survey among stakeholders of research projects in France.
7. Performing interviews with specific stakeholders of research projects in France.
8. Organizing workshops with specific stakeholders of research projects in Poland.
9. Organizing workshops with specific stakeholders of research projects in France.
10. Preparing research results.

Network diagram (presented in Figure 4) explains the sequencing of activities that needs to be applied for the analysed project. In the diagram the red colour represents the project critical path.

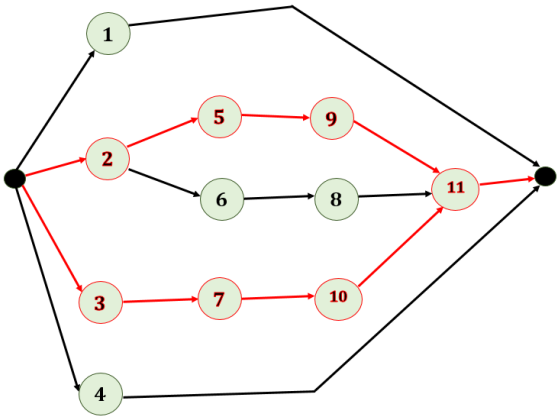


Figure 4: Project Network Diagram.

Gantt Chart for the analysed research project is shown in Figure 5. Meetings (M1-M4) with different stakeholders are also marked in this figure.

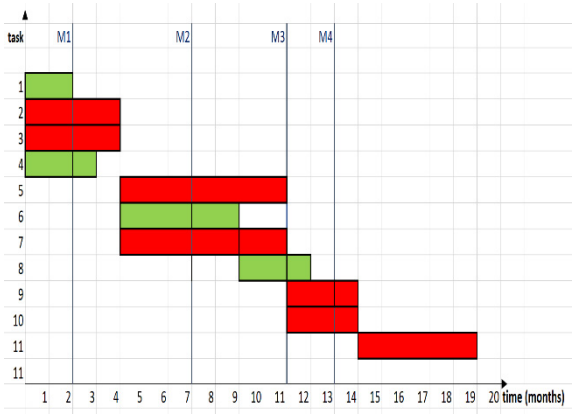


Figure 5: Initial Gantt Chart.

The first important meeting (M1) in our project takes place in the second month. This is a scientific seminar not only for the project team but also for other scientists from the university. At the seminar the concept of the project is presented. From the point of view of scientists participating in the seminar the important outcome of this meeting is building the common understanding of its idea that definitely must be supported by the analysis of the literature. That is why we can state that we should crash activity 4. In traditional project management approach activity 4 will never be crashed if its length is smaller than activity 2 or 3. But in this case we get a greater reward if we shorten activity 4. $B_4^1 = 500\ zl$ (all bounties are presented in Table 2) is the bounty rewarded for completing activity 4 before starting the meeting M1, specified by Project Manager. For scientists activity 4 is much more important than having a tool to manage our project (activity 1) or how many contacts

Table 2: Input data.

V	Pred.	D	C	D^C	C^C	B_i^1	B_i^2	B_i^3	B_i^4
0	∅	0	0	0	0	0	0	0	0
1	{0}	2	2000	1	1000	0	0	0	0
2	{0}	4	500	1	200	0	0	0	0
3	{0}	4	500	1	200	0	0	0	0
4	{0}	3	500	1	200	500	0	0	0
5	{2}	7	5000	1	1500	0	0	0	0
6	{2}	5	2000	2	1000	0	2000	0	0
7	{3}	7	5000	1	1500	0	0	0	0
8	{6}	3	3000	1	1000	0	0	2000	0
9	{5}	3	6000	1	0	0	0	0	7000
10	{7}	3	10000	1	0	0	0	0	0
11	{9, 10}	5	7000	0	0	0	0	0	0
12	{11}	0	0	0	0	0	0	0	0

W	MT
1	2
2	7
3	11
4	13

among stakeholders of research projects we have (activities 2, 3). Moreover if scientists are properly understanding the subject of our project, they will better support our research by answering questions in interviews (activities 6, 8) or filling in survey questionnaires (activities 5, 7). Based on this example we can observe that in some cases it is worth crashing a shorter activity before the meeting, while leaving longer activity uncompleted after the meeting. Such decisions are specific for agile approach. In the example above, a completed activity 4 is shown to the stakeholders (scientists) while other activities, which are not important to the scientists, are not crashed even though they are cheaper to crash than activity 4.

The second meeting (M2) in our project takes place in the seventh month. After three months of performing activities 5, 7 the survey team expects the results from the interviews (activity 6) because they can help to determine the final version of the questionnaire for stakeholders in Poland and France.

Therefore the activity 6 should be shortened to 3 months. Project Manager defined that for completing activity 6 before starting M2 the rewarded bounty is $B_6^2 = 2000\ zl$. Thanks to this operation of shortening activity 6, we not only get a reward in the form of better questionnaires but also the interview team is able to start immediately the next interview with French scientists (activity 8). As a consequence we are then allowed to finish activity 8 before the third meeting (M3) in 11th month. This is a meeting with Rector of the Wroclaw University of Technology to report progress in our project and may give us another reward. Project Manager defined that for completing activity 8 before starting M3 the rewarded bounty is $B_8^3 = 2000\ zl$. The fourth meeting (M4) in our project takes place in the thirteenth month when we have to report progress in our project to NCN (Narodowe Centrum Nauki – National Science Centre).

This is the example of crashing activities in series. Activity 9 cannot be performed until activity 5 is completed. In traditional approach crashing the project is done by crashing the cheapest-to-crash activity situated on the critical path. However, when 5 and 9 have comparable crashing costs, crashing 5 to meet the deadline is preferable, even when the crashing cost for 5 is slightly higher than the crashing cost for 9. Crashing early is important in time-constrained projects as it gives room for the possible last-minute crashing of final activities. That could not be done in the case when these activities are already crashed to their minimum length. In our case we crash activity 9 before the fourth meeting with National Science Centre because of the two reasons:

- activity 5 cannot be shortened any more,
- Project Manager defined that for completing activity 9 before starting meeting M4 the rewarded bounty is $B_9^4 = 7000\ zl$. It is because of the fact that National Science Centre is more interested in the results of

Polish workshops than in results of surveys.

Table 2 presents input data to the model in the analysed case of research project. Based on the project documentation the dependencies between the activities, durations of activities (D) and their costs (C) were determined. Furthermore Project Manager defined: dates of meetings with specific stakeholder of the project (MT), maximum crashing durations for each activity (D^C), crashing cost per time unit for each activity (C^C) and bounties for completing specific activities before meetings planned with different stakeholders ($B_i^1, B_i^2, B_i^3, B_i^4$). In this case there were no budget limits in the span between meetings (M).

According to the data of the project described above, the following model can be built based on Eqs. (1), (3)-(7) in Section 3:

Min cost : $\sum_{i=0}^{12}(C_i^c \cdot x_i - \sum_{j=1}^{4} o_i^j \cdot B_i^j)$ s.t.

$\sum_{i=0}^{12} o_j^i \cdot C_i \leq M_j$, $j = 1,2,3,4$

$x_i \leq D_i^C$, $i = 0,1,\ldots,12$

$y_0 = 0$ and $y_{12} \leq 20$

$y_j - y_i + x_i \geq D_i$, $i = 0,1,\ldots,12$, $j = 1,2,3,4$

$10000 \cdot o_j^i \geq MT_j - y_i$, $i=0,1,\ldots,12$, $j=1,2,3,4$

$10000 \cdot (1 - o_j^i) \geq -(MT_j - y_i)$, $i=0,1,\ldots,12$, $j=1,2,3,4$

Table 3 presents results obtained by solving the given model and Figure 6 shows the updated schedule, including the changes described above. In Figure 6 dashed stroke and light colour mean old tasks and continuous stroke and saturated colour mean updated tasks.

Table 3: Results.

V	x_i	y_i	o_i^1	o_i^2	o_i^3	o_i^4
0	0	0	1	1	1	1
1	0	0	1	1	1	1
2	0	0	0	1	1	1
3	0	0	0	1	1	1
4	1	0	1	1	1	1
5	0	4	0	0	1	1
6	2	4	0	1	1	1
7	0	4	0	0	1	1
8	0	7	0	0	1	1
9	1	11	0	0	0	1
10	0	11	0	0	0	0
11	0	14	0	0	0	0
12	0	19	0	0	0	0

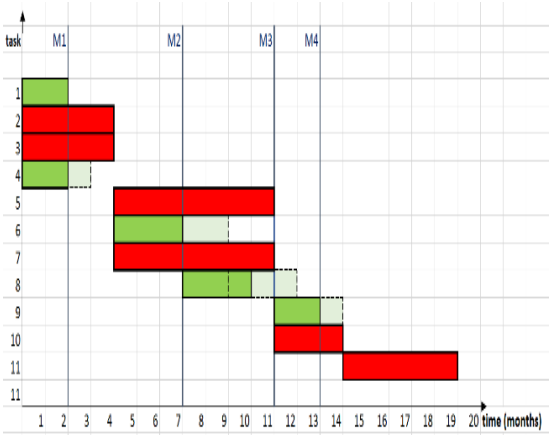


Figure 6: Uploaded Project Network Diagram.

We have seen in the analysed case that the model can render in a systematic manner decisions that make sense from the point of view of satisfying selected project stakeholders during the project realisation but are not the choice that would have been retained by a traditional crashing. This allows for better project planning in an environment where stakeholders stay present throughout project execution and may influence it and its perception. In the analysed case the proposed approach would positively influence the satisfaction of the following stakeholders:

1. Scientists – because of completing activity 4 before starting meeting M1.
2. The survey team – because of completing activity 6 before starting meeting M2.
3. Rector of the Wroclaw University of Technology – because of completing activity 8 before starting meeting M3.
4. National Science Centre – because of completing activity 9 before starting meeting M4.

The proposed model has also its weaknesses. The more input data we have, the more accurate model we get. This can cause problems for the projects where no input would be available. Our future work will focus on the dependencies between the input dataset size and the accuracy of the estimations. Another future work is to examine more complicated relationships between the parameters of our model (e.g. non linear relationships between crashing cost and crashing duration) as well as to investigate the accuracy of our model on projects of different sizes and number of available resources.

5 CONCLUSIONS

Recent years have shown a paradigm shift in project

management approaches. The most important change can be observed in the increasing role of client and other stakeholders in managing projects. Traditionally the client input was important only during planning phase and the acceptance. Nowadays client feedback is important all along the project execution. The changes can be particularly observed in the industries where this feedback can lead to better and more accurate products, such as software development. It is always worth combining both agile and traditional methods of project management. This article presents such an approach. The example from this article shows the project type where this combined approach leads to better results. The method presented in this article helps in managing projects in fast changing environments where the input of the client and stakeholders can shape the initial scope of the project, but where the customer satisfaction is maximized at each stage of project execution.

REFERENCES

Abbasi, G. Y., Mukkatash, A. M., 2011. Crashing PERT networks using mathematical programming, *International Journal of Project Management*, Vol. 9.

Al-Taani R. H., Razali R., 2013. Prioritizing Requirements in Agile Development: A Conceptual Framework, *Procedia Technology Vol. 11*.

Al-Tmeemy S.H., Abdul-Rahman H., Harun Z., 2010. Future criteria of success of building projects in Malaysia*, International Journal of Project*.

Alba E., Software project management with gas, 2007. *Information Sciences 177* (11).

Blumer Y. B., Stauffacher M., Lang D. J., Hayashi K., Uchida S., 2013. Non-technical success factors for bioenergy projects-Learning from a multiple case study in Japan. *Energy Policy*.

Brucker M. N. P., 1999. Resource-constrained project scheduling: Notation, classification, models, and methods, *European Journal of Operational Research 112* (1).

Dohi S. O. T., Nishio Y., 1999. Optimal software release scheduling based on artificial neural networks, *Annals of Software Engineering*.

Elkadi H., 2013. Success and failure factors for e-government projects: A case from Egypt. *Egyptian Informatics Journal*, 14(2).

Freeman R. E, 1984. Strategic Management: a stakeholder approach, Pitman Series in *Business and Public Policy, Harpercollins, College Div*.

Kosztyn Z. T., 2015. Exact algorithm for matrix-based project planning problems, Expert Systems with Applications 42 (9).

Luglio F., Bertazzoni N., 2010. Research management in higher education institutions: A process management experience in Italian Universities, CRIS 2010: Connecting Science with Society - The Role of Research Information in a Knowledge-Based Society - *10th International Conference on Current Research Information Systems*.

Powers L.C., Kerr G., 2009. Project Management and Success in Academic Research. Realworld Systems.

Spundak M., 2014. Mixed agile/traditional project management methodology reality or illusion?, *Procedia – Social and Behavioral Sciences, Vol. 119*.

Wysocki R. K., 2014. Effective Project Management: Traditional, Agile, Extreme. Wiley.

Zou W., Kumaraswamy M., Chung J., Wong J., 2014. Identifying the critical success factors for relationship management in PPP projects. *International Journal of Project Management*, 32(2).

Structuring Guidelines for Web Application Designers
A Meta-model

Anh Do Tuan[1], Isabelle Comyn-Wattiau[1,2] and Samira Si-saïd Cherfi[1]
[1]*CEDRIC-CNAM, Paris, France*
[2]*ESSEC Business School, Cergy-Pontoise, France*
anhdt1@gmail.com, {wattiau, samira.cherfi}@cnam.fr

Keywords: Web Application Design, Guideline, Meta-model, Quality Characteristic, Knowledge Capitalization.

Abstract: Companies develop and maintain complex web sites. Literature provides them with many guidelines for these tasks. However this knowledge is disseminated in many information sources and difficult to apply. This paper is an attempt to address the following research question: How to structure the existing guidelines helping website designers in order to facilitate their application? The contribution is twofold: i) we propose a meta-model allowing a rich representation of these guidelines, ii) we feed this model with several hundreds of guidelines thanks to a deep extraction and structuration. Future research will consist in enriching the UWE (UML-based Web Engineering) method with this knowledge base leading to a quality based approach.

1 INTRODUCTION

Companies develop and maintain complex web sites that allow them to communicate easily and dynamically with their customers, suppliers, partners, etc. In 2008, according to Krigsman, 24% web projects fail to be delivered within budget and 5% were unable to confirm the final cost of their web development project. Moreover, 21% fail to meet stakeholder requirements and nearly a third of web based projects (31%) were not delivered within the agreed timescales (Krigsman, 2008). More recently, a research, conducted by McKinsey and the University of Oxford on more than 5400 IT projects, concluded that 45% of large projects are over budget, 7% are over time and 56% delivered less value than predicted (Bloch, 2013). The reasons vary: unclear objectives, lack of business focus (missing focus), shifting requirements, technical complexity (content issues), unaligned team, lack of skills (skill issues), unrealistic schedule, reactive planning (execution issues) (Bloch, 2013), inconsistent stakeholder demands, and insufficient time or budget (Krigsman, 2008).

Web sites and web applications are in fact software applications. In this sense, the classical application methodologies may be used manually or with the help of computer aided software engineering (CASE) tools. However, the very specific nature of these applications led to the proposition of more dedicated approaches. Indeed, during the two last decades, research in Web Engineering brought a rich contribution composed of methods and techniques to support Web applications development. These methods such as UWE, WebML, or others are generally founded on a model-driven development paradigm, and provide models and transformation rules to handle several web applications' aspects such as data, navigation, interaction, and presentation. However and despite the research and the tooling efforts, very few developers adopt these methods and many continue to apply ad-hoc practices.

The main reason is that these approaches suffer from lack of guidance. Even if web application designers refer to these approaches, they do not have sufficient knowledge and help in implementing them efficiently. As a consequence, the resulting applications are neither user-friendly nor easy to maintain.

We argue that the current approaches are well structured. However they need to be enriched with guidelines helping designers in the numerous decisions they have to make during the web application development. Therefore, we have collected the different sets of guidelines proposed in the literature and organized them along different dimensions. In particular, this structure allows us to link the guidelines with the quality objectives (maintainability, performance, functionality, security, etc.) and with the relevant steps of the web application design (content design, navigation design and presentation design).

Tuan, A., Comyn-Wattiau, I. and Cherfi, S.
Structuring Guidelines for Web Application Designers - A Meta-model.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 327-335
ISBN: 978-989-758-187-8

This article is organized as follows. Section 2 describes how we collected and selected the guidelines, and a short experiment we conducted on how methods and guidelines are followed in websites construction. Based on the survey conclusions, Section 3 motivates and describes our research question. Section 4 describes the meta-model we propose in order to represent the guidelines in a useful way. Section 5 analyses the set of resulting guidelines. Section 6 is dedicated to related works on guidelines. Finally, the last section concludes and sketches future research directions.

2 AN EXPERIMENT ON METHODS AND GUIDELINES USAGE

Before defining the research question, we performed a quick inventory on how well web design best practices and guidelines are followed by existing websites. The objective was i) to analyze whether existing practices and guides are used and ii) identify how to facilitate their adoption and hence avoid ad hoc approaches. Thus, we first collected 475 guidelines from several sources and confronted them with three websites: the web site of our university department (deptinfo.cnam.fr), the website of a French newspaper (lemonde.fr) and a well-known e-commerce web site (amazon.fr). We first describe briefly the collected guidelines and then their verification on the three websites.

2.1 Collecting the Guidelines

World Wide Web Consortium (W3C) is the main international standards organization for the World Wide Web. This consortium puts together around 400 organizations. They developed Web Content Accessibility Guidelines (WCAG) with the goal of proposing a single shared standard for web content accessibility that meets the needs of individuals, organizations, and governments (Web Accessibility Initiative). Two versions of WCAG were published until now. The first one was introduced in 1999. It contains 14 large guidelines. Each main guideline is composed of atomic guidelines addressing the same topic. The second version was published in 2008. It contains 12 guidelines organized into four categories, targeting four desirable characteristics of websites: perceivable, operable, understandable, and robust.

WCAG defines three levels of conformance, respectively A, AA and AAA. Some of the related guidelines could be automatically checked whereas others require manual checking. Authors in (Trulock, 2008) conducted a case study on Irish websites showing that web designers are aware of web accessibility but they concentrate their efforts on ensuring validation of automatically controlled checkpoints and ignore those requiring additional manual testing.

The guidelines of WCAG focus only on accessibility. Thus, we collected other guidelines which address all the characteristics of web site quality. The literature contains guidelines for specific web sites (for children for instance) as well as rules available for all sites.

2.1.1 Identifying the Relevant Sources

For collecting guidelines from literature effectively, we use some keywords when searching, such as “website guideline”, “guideline for website”, “guideline security web application” in title and content of document, from main electronic libraries and databases in computer science: IEEE Xplore, Springer, ScienceDirect, ACM, and DBLP. As an example, based on the keywords “web” and “guideline”, we have 1273 results from IEEE, 273 results from ScienceDirect and 168 results from DBLP. With Springer and ACM, we have much more results in many domains, so we had to refine the results and choose results with high relevance (as computed by the search engines). Then we defined inclusion criteria for selecting sources (primary studies) and rejecting the other ones. The inclusion criteria are presented in the table below (Table 1).

Table 1: Inclusion criteria.

Criterion	Description
C1	The study focuses on guideline definition for web sites
C2	The study mentions quality characteristics of web sites
C3	The paper is recent, i.e. published since 2000
C4	The paper proposes original guidelines (does not only mention guidelines from other studies)

We found several guideline lists published since 2000. However, these documents are sparse and address many domains. One objective is to gather them, categorize, and model guidelines. Thus they will be more usable for supporting web application developers. Some guidelines are general and others are dedicated to specific domains: education, international, or for particular ages (children or

seniors). As an illustration, the guidelines of AgeLight Company (AgeLight LCC, 2001) are divided in six categories: layout and style, color, text, general usability testing, accessibility and disabilities, user customization. Web sites for old people are the research object of a number of studies (Xie, 2011) (Sun, 2010). Meloncon et al., in contrast, concentrated on guidelines for children (Meloncon et al., 2010). Maguire focused on e-commerce international sites (Maguire, 2011). Some papers focused on the characteristics of quality directly, such as (Chiuchi, 2011) which targeted portability and efficiency. (Radosav, 2011) capitalizes on the 14 guidelines from WCAG, so we did not collect them. Finally, we took into account fourteen sources. Their analysis is described below.

2.1.2 Extracting the Appropriate Guidelines

Our systematic search followed by a scan of sources allowed us to exhibit fourteen papers containing relevant guidelines. The next step consisted in studying all the guidelines and selecting the helpful guidelines. In each source of guidelines, we found some obsolete guidelines or some recommendations which were out of our scope. For example, in (U.S. Department of Health and Human Services, 2006), guidelines in the last part (part 18), such as "Use an iterative design approach" or "Solicit test participants' comments" were not selected, since they are too general or dedicated to testing. So we eliminated them from the list.

We found 14 sources with 475 guidelines (an excerpt is listed at Annex 1). The number of guidelines of each source is presented in Table 2. In some cases, we split some guidelines, hence the number of selected guidelines may be higher than the number of guidelines proposed in the paper.

Some sources propose general guidelines. Others are more specific. For example (Bargas-Avila, 2010) concentrates on web forms or (Chiuchi, 2011) focuses on portability and efficiency.

Some guidelines are too complex, so we had to divide them into two parts or more. For example the guideline for images in (Chiuchi, 2011) is separated into two atomic guidelines: "The preferred use of JPEG and GIF images" and "The resolution of image should be set correctly inside the tags".

2.2 Analyzing the Guidelines Usage

To analyze how well the guidelines are applied in practice, we defined four levels namely: Yes, No, Partial and NN. Yes means that the site satisfies completely the guideline, No means that this site does not satisfy it, Partial means that this site partially meets the guideline and NN means that "We don't know", since either the guideline cannot be applied to the site or we don't have enough information. The result is synthesized at Figure 1. Each guideline obtains the grade 1, 0.5, 0 point for Yes, Partial and No respectively. After applying all guidelines to the three websites, each guideline obtains a grade between 3 and zero or is equal to NN. Thus, 206 guidelines are verified on the three selected sites (totalizing 3 points). 33 guidelines reach 2.5, 46 guidelines obtain 2 points. 60 guidelines obtain between 0.5 and 1.5. 47 guidelines obtain 0, meaning that they are not respected on the three selected sites.

Table 2: Source, number and scope of guidelines.

Source	Proposed guidelines	Selected guidelines	Scope
(AgeLight LCC, 2001)	53	35	General
(Bargas-Avila, 2010)	20	20	General but concentrating on web forms
(Xie, 2011)	7	10	Old people / medical information
(Chiuchi, 2011)	17	15	General / focusing portability and efficiency
(Carnegie Mellon University)	7	8	University
(U.S. Department of Health and Human Services, 2006)	196	209	General
(Leuthold, 2008)	9	9	Blind people
(Lokman, 2009)	13	14	General
(Maguire, 2011)	20	8	International site
(Meloncon, 2010)	21	11	Children
(Microsoft Developer Network)	50	49	General
(Ministry of Community and Social Services of Ontario, 2012)	11	11	General
(Ozok, 2004)	20	20	General
(Sun, 2010)	31	31	Old people

But let us remind that 3 guidelines are dedicated to international or children sites, and thus are not required in the three tested web sites. Besides them, there are 83 guidelines obtaining the NN value. For guidelines which have NN value, many of them are related to the security aspects. To check if they are fulfilled, we require the admin authority, so we cannot conclude about these guidelines.

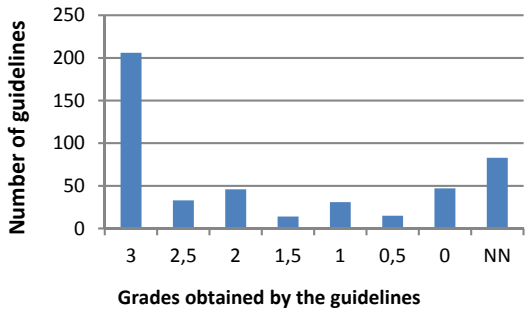


Figure 1: The distribution of guideline grades.

As an illustration, the guideline G115 "considering both levels: 'high' and 'low' of cultural context for satisfying both viewpoints" or G176 "Limit navigational topics" are not relevant for the three web sites. Others may be irrelevant, such as G217 "Inform users of long download times" or G247 "Limit homepage length" since we had high speed connection for our tests.

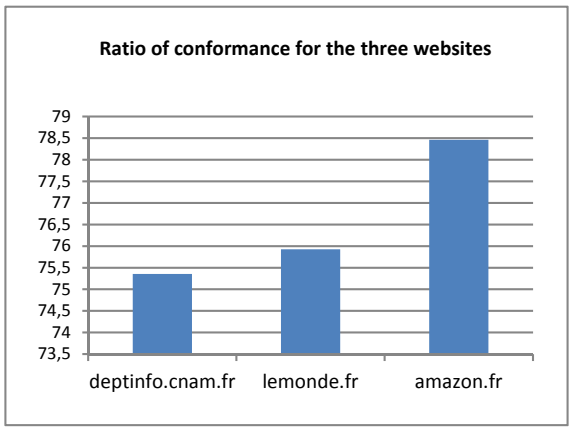


Figure 2: Results for the three websites.

Figure 2 compares the scores obtained by the three websites if we consider the rule: the more guidelines the web site complies with, the better score it obtains. deptinfo.cnam.fr obtains the score of 264.5 while lemonde.fr obtains 287 points. Finally, amazon.fr is the best one with score of 300.5. However, deptinfo.cnam.fr has 124 NN guidelines, whereas lemonde.fr has only 97 NN guidelines and amazon.fr has 92 NN guidelines, so if we compare ratios, deptinfo.cnam.fr achieves 75.3%, lemonde.fr 75.9% and amazon.fr is the highest with 78.5%.

These figures show that either these guidelines are not considered as references or these websites still face quality problems. As an example, let us mention G345 "Provide auto-tabbing functionality" for increasing users' convenience and G362 "Using photographs of people" for increasing users' reliability. The three websites are not aligned with these two guidelines. That means that these guidelines which were validated through complex processes are not sufficiently known by web site designers.

3 RESEARCH QUESTION

From the mid of 1990s, methods and approaches have been created for helping developers to build web applications more easily and constructively. The Object-Oriented Hypermedia Design Method (OOHDM) was one of the first methods proposing a rigorous process from requirements elicitation to implementation including navigational and interface design (Schwabe, 1995). The method relies on object-oriented principle and proposes notation mainly derived from UML. The transition from models to specification is not supported and thus requires a considerable effort.

The Web Modelling Language (WebML) is a model driven web engineering method dedicated to data-intensive web applications (Ceri, 2000). WebML is one of the most used web engineering methodologies. It is supported by a development framework, Ratio5 (Acerbis, 2005) that is fully integrated to the Eclipse framework. Several extensions of the first version have been proposed offering a rich modelling approach for developers. However, the method relying very few on standards, it led to a proliferation of proprietary notations increasing the method complexity.

The UML-based Web Engineering (UWE) methodology (Hennicker, 2000) is a model-driven Web Engineering approach. It relies heavily on UML and is extensively related to standards. The model driven orientation allows generating platform specific implementation through dedicated transformation rules. Model driven approaches are based on four levels of abstraction: the computer independent model (CIM), the platform independent model (PIM), the platform specific model (PSM), and the code. Some methods address only the CIM level, other methods focus on PIM level. In the same way, some

methods deal with the transformation of CIM to PIM (e.g. NDT, OOWS), others address the transformation of PIM to PSM (e.g. WebML, UWE) and others incorporate the transformation of PSM to code (e.g. OOHDM, UWE) (Aragon, 2013). Even if these methods offer a real support, they are still not used by practitioners probably since they are complex and they do not provide designers with sufficient guidance.

We argue that most methods do not provide their users with sufficient guidance in the design and development process. Either in the same approaches or in other sources, researchers propose many guidelines in order to help designers and developers. These guidelines may be very helpful to support them.

Thus the research question we address in this paper may be defined as follows: "How to structure all the existing guidelines helping website designers to understand and apply them?" To answer this question the experiment presented in Section 2 helped us to elicit the main characteristics of these guidelines. We then defined a meta-model allowing us to represent this knowledge. Finally we categorized the selected guidelines based on our meta-model. This categorization aims to facilitate their reuse.

4 GUIDELINE CAPITALIZATION: A MODEL-BASED APPROACH

In the literature, we find different ways to describe guidelines: in (Chiuchi, 2011), they are represented by three attributes: Category, Name and Content. Meanwhile in (Ekberg, 2010) a guideline has three parts: design/application solutions, objective and description. We argue that this descriptive information is not sufficient to facilitate the reuse of guidelines by web application designers. In particular, the latter must find easily the guidelines using different criteria. For example, in case of designing a web application for blind people: which recommendations do they have to take into account? If developers want mainly to facilitate the maintainability of the web application: which guidelines aim at this objective? Etc.

We first propose a model helping capitalizing and structuring the guidelines. The meta-model is depicted at Figure 3.

Following the general description of patterns for decision processes (Harrison, 2007), we propose to link each guideline with the following categories:

- The source where the guideline was found,
- The quality characteristics and sub-characteristics that the guideline addresses,

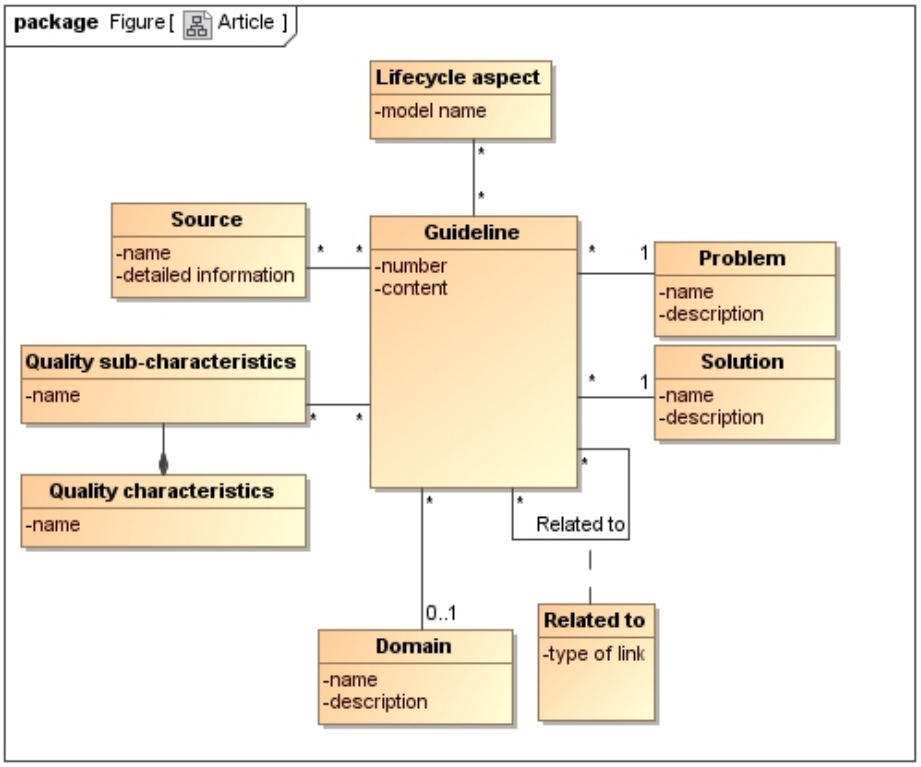


Figure 3: The meta-model of guidelines.

- The problem it aims to solve,
- The solution proposed,
- The particular domain concerned if any,
- The lifecycle aspect, meaning which web application model (content model, navigation model, presentation model) it deals with.

This structure will constitute a knowledge base for automatic reuse through a web application design tool. The meta-model is represented as a UML class diagram at Figure 3. The *related to* relation between guidelines allows us to represent potential links between guidelines. Thus the attribute type of link may take the values "in contradiction with", "specializes" or "similar to".

Each guideline solves a problem; however several guidelines may tackle the same problem. The solution of the guideline describes the rules to be applied. As explained above, in our process, we split some guidelines such that each resulting guideline recommends one and only one solution. The domain may be general or it may be a specific one. The quality characteristics (functional suitability, performance/efficiency, compatibility, usability, reliability, security, maintainability, portability) and sub-characteristics refer to ISO25010 for software quality. For space reasons we may not list all of them. Some guidelines are common to several sources, hence the multiplicity of the relation here is many-to-many. Finally, the lifecycle aspect consists of three elements: Content, Navigation, and Presentation.

In order to illustrate, let us describe the guideline G37: "For body copy, the recommended faces for the web, in order of preference, are Verdana, Arial and Helvetica. The browser should use Verdana first; if it is not available, use Arial and then Helvetica. If none are available, use another Sans serif font".

Number: #37
Content**:** For body copy, the recommended faces for the web, in order of preference, are Verdana, Arial and Helvetica. The browser should use Verdana first; if it is not available, use Arial and then Helvetica. If none are available, use another Sans serif font.
Problem: Choosing appropriate font for a website
Domain: web for university (even if it can also apply to other types of site)
Lifecycle aspect: Presentation
Quality Sub-characteristics: User interface aesthetics
Quality Characteristic: Usability
Solution: Choose Sans serif font, namely Verdana, Arial and Helvetica.
Source: (Carnegie Mellon University)

5 GUIDELINES ANALYSIS

In this section, we provide the reader with an analysis of the guidelines according to the different dimensions of our meta-model. Let us remind that our selection process led to the constitution of a set of 475 guidelines (the guidelines can be found at http://deptinfo.cnam.fr/~wattiaui/Guidelines.html).

If we analyze them from the lifecycle dimension (Content/ Navigation/ Presentation), we counted 203 guidelines for Presentation, 291 guidelines for Content and only 40 guidelines for Navigation. Some guidelines address more than one model. Hence the total exceeds 475 (Figure 4).

The 475 guidelines were mapped with quality sub-characteristics. Some guidelines are mapped with several sub-characteristics. The characteristic Usability, with sub-characteristics Operability and User interface aesthetics is the most involved one. It is easy to explain since many papers address interface aspects (User interface aesthetics) and aim to build easy-to-use interfaces (Operability).

Many guidelines are about font (G37, G42, G49, G50, etc.) and color (G6, G8, G39, G41, G86, G185, G186, etc.) of websites. White is the color which is not recommended (G9, G39, G189, etc.).

We can detect some contradictory guidelines, since some guidelines aim at different goals. In the guidelines of a university (Carnegie Mellon University) the documents should be opened in new windows (G35), probably for legal responsibilities. It is opposite to guideline G101 (AgeLight LCC, 2001) which recommends not to open external links in new windows, since it can cause user distracting.

Guideline G37 recommends using only Sans-serif font, but meanwhile G85 accepts serif font in web site for printing.

Some guidelines are dedicated to different types of users, but finally they have same contents. As an illustration, Sun et al. (Sun, 2010) focused on website for old people; meanwhile Meloncon et al. (Meloncon, 2010) concentrated on web applications for children. Old people and children are two types of users which have some specific characteristics in comparison with others (e.g. not being able to understand complex content).

The guideline about Security of web applications in MSDN of Microsoft (Microsoft Developer Network) contains about 50 sections. Many of them address Integrity (prevent unauthorized access) (in 38 sections) and Confidentiality (data are accessible only to those authorized) (in 18 sections). This is due to the fact that Integrity and Confidentiality are important for web applications which are designed for many

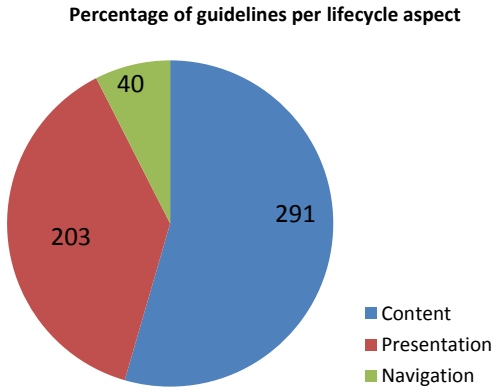


Figure 4: Percentage of guidelines per lifecycle aspect.

kinds of users and also are the targets of attacks.

Among the eight quality characteristics, Compatibility is not mentioned at all, since guidelines focus on the site itself, and not on the relation of the site with other sites or other applications (scope of Compatibility).

6 RELATED WORKS

In this section, we synthesize the literature on guidelines for web site design. We organized this state of the art in two categories: first the approaches which propose guidelines and, second, the approaches which involve such guidelines.

6.1 Design for Guidelines

One of the most famous works delivering guidelines for web site design is Web Accessibility Initiative (WAI) of W3C (WAI, 2008). It is a collection of standards, guidelines, and techniques for making accessible products in four categories: websites, authoring tools, browsers, and web applications. Each category has a bunch of guidelines for constructing web design and for improving accessibility. Other sources of guidelines were listed in the paper and enrich considerably the W3C recommendations.

Khlaisang is an example of research illustrating how these guidelines may be either validated or elicited (Khlaisang, 2015). The author developed user interface guidelines and a prototype for evaluating educational service websites. Based on source sites of Thailand Cyber University Project (TCU), he studied the use of sites, the website structure, the user interface design and conducted usability tests of the site. Resulting from these experiments, he presented a model of suitable website for TCU service. Starting from this website, model, he designed and developed a prototype of site. The paper also mentions similar approaches.

6.2 Design by Guidelines

Besides works creating guidelines, other works used existing guidelines for proposing ways to improve quality of websites.

Leuthold et al. (Leuthold et al., 2008) designed enhanced text user interfaces for blind Internet users. Starting from the guidelines of web content accessibility guidelines (WCAG), they proposed enhanced text user interface (ETI) helping blind users in spending less time to complete tasks, making fewer mistakes and expressing greater satisfaction when surfing the website. This system contains nine guidelines. For blind users, this system is more usable than normal GUI.

Another work building on WCAG guidelines is (Sloan, 2006). Using e-learning as an example, they propose a framework that guides web authors and policy makers in addressing accessibility at a higher level, by defining the context in which a Web resource will be used and considering how new alternatives may be combined to enhance the accessibility of the web site.

After a brief description of the 14 guidelines of WCAG (version 1), Radosav et al. discussed the choice of colours for adjusted web design (Radosav, 2011). They classified colour into several groups and concluded that colours, which cannot be differentiated by people with colour discrimination disability, should not be placed next to each other.

For space reasons, we cannot provide a more detailed literature review. As a conclusion, research in this field is prolific and aims at i) proposing guidelines for web site designers, ii) enriching existing ones, iii) implementing guidelines into more comprehensives approaches, iv) evaluating guidelines through experiments. To the best of our knowledge, we did not find any paper proposing a meta-model allowing us to put together the different guidelines as a first step for their reuse in an automatic way.

7 CONCLUSION AND FUTURE RESEARCH

The companies grasp the importance of having usable and efficient web applications. Thus, their development and maintenance is of high importance. The academic literature on the subject contains hundreds of guidelines aiming at helping web site

designers. The research question we addressed in this paper may be expressed as follows: How to structure the existing guidelines helping website designers in order to facilitate their application? As a first contribution, we defined a meta-model allowing us to describe each guideline with six dimensions: the problem it addresses, the solution it proposes, the lifecycle aspect it deals with, the target quality characteristics, the source it comes from, the potential links (similarity, contradiction, specialization) with other guidelines. Our search and selection process allowed us to define 475 such guidelines and to feed our meta-model with them. This required the mapping of them with the relevant quality sub-characteristics. As a first evaluation of these guidelines, we checked whether they were compliant with three very different web sites.

This research suffers from some limitations. Thus, it is rather easy to check the contradiction between guidelines attached to the same quality characteristics and/or sub characteristics. However, contradictions may also occur between guidelines associated with different quality characteristics. Moreover, some guidelines may become obsolete due to new technical opportunities. It is not easy to ensure an easy update of guidelines.

Future research will explore three directions: first the definition of a grammar for expressing problem and solution components of guidelines; second, the implementation of these guidelines in a CASE tool implementing UWE web application design method; third, a validation of the approach through an experiment with web site designers, in order to evaluate how the guidelines help them when using the CASE tool.

REFERENCES

Acerbis, R., Bongio, A., Brambilla, M., & Butti, S. (2007). Webratio 5: *An eclipse-based case tool for engineering web applications. In Web Engineering* (pp. 501-505). Springer Berlin Heidelberg.

AgeLight LCC, 2001. Interface design guidelines for users of all ages. Technical report, 2001.

Aragon, G., Escalona, M.J., Lang, M., Hilera J., 2013. *An Analysis of Model-Driven Web Engineering Methodologies*. International Journal of Innovative Computing, Information and Control Volume 9, Number 1, January 2013.

Bargas-Avila, J.A., Brenzikofer, O., Roth, S.P., Tuch, A.N., Orsini, S., Opwis, K., 2010. *Simple but Crucial User Interfaces in the World Wide Web: Introducing 20 Guidelines for Usable Web Form Design*. In User Interfaces, Rita Matrai (Ed.), InTech, 2010.

Bloch, M., Blumberg, S., Laartz, J., 2013. *Delivering large-scale IT projects on time, on budget, and on value*. McKinsey on Finance Number 45, Winter 2013, pp 28 – 35.

Carnegie Mellon University. *Web guidelines*. http://www.cmu.edu/marcom/brand-guidelines/print-web-products/web/index.html.

Ceri, S., Fraternali, P., & Bongio, A. (2000). *Web Modeling Language (WebML): a modeling language for designing Web sites*. Computer Networks, 33(1), 137-157.

Chiuchi, C.A., de Souza, R.C.G., Santos, A.B., Valêncio, C.R., 2011. *Efficiency and portability: guidelines to develop websites*. Software Engineering and Knowledge Engineering 2011: Miami Beach, USA, pp. 37 – 41.

Ekberg, J., Ericson, L., Timpka, T., Eriksson, H., Nordfeldt, S., Hanberger, L., Ludvigsson, J., 2010. *Web 2.0 Systems Supporting Childhood Chronic Disease Management: Design Guidelines Based on Information Behaviour and Social Learning Theories*. Journal of Medical Systems April 2010, Volume 34, Issue 2, pp 107-117.

Harrison, N.B., Avgeriou, P., Zdun, U., 2007. *Using Patterns to Capture Architectural Decisions*. IEEE Software, Vol 24(4), 2007, pp. 38-45.

Hennicker, R., & Koch, N. (2000). *A UML-based methodology for hypermedia design*. In ≪ UML≫ 2000—The Unified Modeling Language (pp. 410-424). Springer Berlin Heidelberg.

Khlaisang, J., 2015. *Research-based Guidelines for Evaluating Educational Service Website: Case Study of Thailand Cyber University Project*. Procedia - Social and Behavioral Sciences, Volume 174, February 2015, pp. 751-758.

Krigsman, M., 2008. *Research: 25 percent of web projects fail*. http://www.zdnet.com/article/research-25-percent-of-web-projects-fail/, May 2008.

Leuthold, S., Bargas-Avila, J.A., Opwis, K., 2008. *Beyond web content accessibility guidelines: Design of enhanced text user interfaces for blind internet users*. International Journal of Human Computer Studies, Volume 66, Number 4, April 2008, pp. 257-270.

Lokman, A.M., Noor, N.L., Nagamachi, M., 2009. *ExpertKanseiWeb: A Tool to Design Kansei Website*. Enterprise Information Systems, 11th International Con-ference, ICEIS 2009, Milan, Italy, May 6-10, 2009: pp. 894-905.

Microsoft Developer Network. *Chapter 4 - design guidelines for secure web applications*. https://msdn.microsoft.com/en-us/library/ff648647.aspx .

Ministry of Community and Social Services of Ontario, 2012. *Making your website more accessible*. Queen's Printer for Ontario, 2012. http://www.mcss.gov.on.ca/en/mcss/publications/accessON/accessible_websites/toc.aspx

Ozok, A., Salvendy, G., 2004. *Twenty guidelines for the design of Web-based interfaces with consistent language*. Computers in Human Behavior. Vol. 20(2), 149-161 (2004).

Maguire, M., 2011. *Guidelines on Website Design and Colour Selection for International Acceptance*. HCI International 2011, pp. 162–171, 2011.

Meloncon, L., Haynes, E., Varelmann, M., Groh, L., 2010.

Building a Playground: General Guidelines for Creating Educational Web Sites for Children. Technical Communication, 57(4):398–415, 2010.

Radosav, D., Karuovic, D., Markoski, B., Ivankovic, Z., 2011. *Guidelines on accessible web portal design*. 2011 *IEEE 12th International Symposium on Computational Intelligence and Informatics* (CINTI), 21-22 Nov. 2011, pp. 297 – 302.

Sloan, D., Heath, A., Hamilton, F., Kelly, B., Petrie, H., Phipps, L., 2006. *Contextual web accessibility, maximizing the benefit of accessibility guidelines*. Proceeding W4A '06 Proceedings of the 2006 international cross-disciplinary workshop on Web accessibility (W4A): Building the mobile web: rediscovering accessibility? Pages 121-131.

Sun, Z., Zhao, Y., 2010. *The preliminary construction of accessibility design guidelines of learning website for old people*. International Workshop on Education Technology and Computer Science, pages 612–615, 2010.

Schwabe, D., & Rossi, G. (1995). *The object-oriented hypermedia design model*. Communications of the ACM, 38(8), 45-46.

Trulock, V., & Hetherington, R. (2008). *Assessing the Progress of Implementing Web Accessibility-An Irish Case Study*. In ICEIS (5) (pp. 105-111).

U.S. Department of Health and Human Services, U.S. General Services Administration, 2006. *Research-Based Web Design & Usability Guidelines*.

Web Accessibility Initiative (WAI), 2008. *Web Content Accessibility Guidelines (WCAG) Overview*. http:// www. w3.org/WAI/intro/wcag.

Xie, B., Watkins, I., Huang, M., 2011. *Making web-based multimedia health tutorials senior-friendly: design and training guidelines*. iConference 2011: Seattle, Washington, USA, February 8-11, 2011, pp. 230-237.

APPENDIX

Annex 1: Excerpt from the set of guidelines.

Number	Content	Sub-characteristics	Source	Related to other guidelines	Lifecycle	Specific
1	Left justified text, text line should not be long	User interface aethestics	[8]	Similar to G70	Presentation	Old people
6	Support users flexible operations (adjustable font size, color conversion)	Accessibility	[8]		Presentation	Old people
7	Ensure links change color after visit	Operability User error protection	[8]	Similar to G63	Presentation	Old people
20	Provide a site-map	Operability	[8]		Navigation	Old people
21	Search engine should have to check and correct misspelled function	Operability	[8]		Content	Old people
30	Consider page download speed - create 'small' pages	Time behaviour	[8]		Content	Old people
31	Do not require 'double clicks'	Accessibility	[8]		Presentation	Old people
32	All images should be JPGs, GIFs or PNGs. JPGs are used for photos. Graphics should use GIF or PNG formats	Functional appropriateness	[15]		Content	University
33	Images have a resolution of 72 dpi and are in either RGB or indexed color modes	Adaptability	[15]		Content	University
36	Links should be relevant text. Do not link words like "here" "this page" etc.	Operability	[15]		Content	University
48	Main background color should be brown, not light blue	User interface aethestics	[17]		Presentation	
55	Including hyperlinks within longer pages so viewers can "jump" with a single click	Accessibility	[6]		Navigation	
76	Should not use exceptionally bright, fluorescence or vibrant colors	User interface aethestics	[6]		Presentation	
95	Archive old articles, while maintaining the actual page URL	Functional completeness	[6]		Content	
99	Trying to link to sites at the highest possible level, in the case "page not found"	Fault tolerance	[6]		Navigation	
103	If using tables, provide an alternate text-only version of page	Replaceability Fault tolerance	[6]		Content	
106	Site should have multi language versions	Operability	[10]		Content	International site
110	Getting the spelling right for the correct market	Functional correctness	[10]		Content	International site
117	Should build auto response service informing that they will receive a full reply within 24 or 48 hours	Availability	[10]		Content	International site
191	Use concrete words, active verbs, and concise sentence structure	Learnability	[9]	Similar to G25	Content	Children site
246	Ensure the homepage looks like a homepage	Appropriateness recognizability	[13]		Content Presentation	

Improving the Specification and Analysis of Privacy Policies
The RSLingo4Privacy Approach

Alberto Rodrigues da Silva[1], João Caramujo[1], Shaghayegh Monfared[1], Pavel Calado[1] and Travis Breaux[2]
[1]*INESC-ID, Instituto Superior Técnico, Universidade de Lisboa, Lisbon, Portugal*
[2]*Institute for Software Research, Carnegie Mellon University, Pittsburgh, U.S.A.*
{alberto.silva, pavel.calado, joao.caramujo, shaghayegh.monfared}@tecnico.ulisboa.pt, breaux@cs.cmu.edu

Keywords: Privacy Policy, Privacy-aware Specific Language, Requirements Specification, Quality of Requirements.

Abstract: The common operation of popular web and mobile information systems involves the collection and retention of personal information and sensitive information about their users. This information needs to remain private and each system should show a privacy policy that describes in-depth how the users' information is managed and disclosed. However, the lack of a clear understanding and of a precise mechanism to enforce the statements described in the policy can constraint the development and adoption of these requirements. RSLingo4Privacy is a multi-language approach that intends to improve the specification and analysis of such policies, and which includes several processes with respective tools, namely: (P1) automatic classification and extraction of statements and text snippets from original policies into equivalent and logically consistent specifications (based on a privacy-aware specific language); (P2) visualization and authoring these statements in a consistent and rigorous way based on that privacy-aware specific language; (P3) automatic analysis and validation of the quality of these specifications; and finally (P4) policies (re)publishing. This paper presents and discusses the first two processes (P1 and P2). Despite having been evaluated against the policies of the most popular systems, for the sake of briefness, we just consider the Facebook policy for supporting the presentation and discussion of current results of the proposed approach.

1 INTRODUCTION

Web and mobile information systems increasingly leverage user data that is collected from multiple sources without a clear understanding of data provenance or the privacy requirements that should follow this data. These systems are based on multi-tier platforms in which each "tier" may be owned and operated by a different party, such as cellular and wireless network providers, mobile and desktop operating system manufacturers, and mobile or web application developers. In addition, user services developed on these tiers are abstracted into platforms to be extensible by other developers, such as Google Maps and the Facebook and LinkedIn social networking platforms. Application marketplaces, such as Amazon Appstore, Google Play and iTunes, have also emerged to provide small developers increased access to customers, thus lowering the barrier to entry and increasing the risk of misusing personal information by inexperienced developers or small companies. Therefore, platform and application developers bear increased, shared responsibility to protect user data as they integrate their services into multi-tier ecosystems.

For example in Canada, Europe and the United States, privacy policies, also called privacy notices (or just "policies" for simplicity), have served as contracts between users and their service providers and, in the U.S., these policies are often the sole means to enforce accountability (Breaux and Baumer, 2011). In particular, Google has been found to re-purpose user data across their services in ways that violated earlier versions of their privacy policy (Farrell, 2011); and Facebook's third-party apps were found to transfer Facebook user data to advertisers in violation of Facebook's platform policies (Steel and Fowler, 2010). Given the pressure to post privacy policies and the pressure to keep policies honest, companies need tools to align their policies and practices. In this respect, we believe developers need tools to better specify their privacy policies at a requirements and architectural-level of abstraction (i.e., denoting the actors, data types and including restrictions on what data may be collected, how it may be used, to whom it may be transferred and for

Silva, A., Caramujo, J., Monfared, S., Calado, P. and Breaux, T.
Improving the Specification and Analysis of Privacy Policies - The RSLingo4Privacy Approach.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 336-347
ISBN: 978-989-758-187-8

what purposes) and that privacy policies only present a subset of this view to the general public. The challenge for these companies is ensuring that developer intentions at different tiers are consistent with privacy requirements across the entire ecosystem. To this end, we conducted a series of studies to formalize a set of privacy-relevant requirements captured from privacy policies.

On the other hand, Requirements Engineering (RE) intends to provide a shared vision and understanding of the system to be developed between business and technical stakeholders (Pohl, 2010; Sommerville and Sawyer, 1997; Robertson, 2006). The adverse consequences of disregarding the importance of the early activities covered by RE are well-known (Emam and Koru, 2008; Davis, 2005). A privacy policy is a technical document that states the multiple privacy-related requirements that a system should satisfy. These requirements are usually defined as ad-hoc natural language statements. Natural language is flexible, universal, and humans are proficient at using it to communicate. Natural language has minimal adoption resistance as a requirements documentation technique (Pohl, 2010; Robertson, 2006). However, although it is the most common and preferred form of requirements representation (Kovitz, 1998), it also exhibits some intrinsic characteristics that often present themselves as the root cause of quality problems, such as incorrectness, inconsistency or incompleteness (Pohl, 2010; Robertson, 2006; Silva, 2014).

The main objective of this research is to improve the understanding and quality of privacy policies by providing a set of languages and tools to align those policies with their practices, namely by introducing a privacy requirements specification approach into the regular software development process that would allow to align multi-party expectations across multi-tier applications. The relevance of this approach, called RSLingo4Privacy, is demonstrated through the analysis and evaluation of real world privacy policies, namely those posted by the most popular web sites. The results of this research is of paramount relevance and impact both to the industrial as well academic communities by promoting a further rigor related the specification and analysis of privacy requirements and consequently by helping developers to avoid the referred inconsistency and better design and implement their systems.

This paper is structured in seven sections. Section 2 introduces the background underlying this research. Section 3 overviews the RSLingo4Privacy approach. Sections 4 and 5 detail two of the key processes included in this approach, respectively, (P1) automatic classification and extraction of statements and text snippets from original policies into equivalent and logically consistent specifications (based on a privacy-aware specific language); and (P2) visualization and authoring these statements in a consistent and rigorous way based on that privacy-aware specific language. Section 6 discusses the related work. Finally, Section 7 presents the conclusion and ideas for future work.

2 BACKGROUND

This section briefly introduces the background of this research, namely introduces the RSLingo and Eddy research projects, which have contributed for the proposed RSLingo4Privacy approach.

2.1 RSLingo and RSL-IL4Privacy

RSLingo is a general approach for the rigorous specification of software requirements that uses lightweight Natural Language Processing (NLP) techniques to (partially) translate informal requirements – originally stated by business stakeholders in unconstrained natural language – into a rigorous representation provided by a language specifically designed for RE. The name RSLingo stems from the paronomasia on "RSL" and "Lingo" (Ferreira and Silva, 2012). On one hand, "RSL" (Requirements Specification Language) emphasizes the purpose of formally specifying requirements. The language that serves this purpose is RSL-IL, in which "IL" stands for Intermediate Language (Ferreira and Silva, 2013). On the other hand, "Lingo" expresses that its design has roots in natural language, which are encoded in linguistic patterns used during by the information extraction process (Bird et al., 2009; Cunningham, 2006; Ferreira and Silva, 2013a) that automates the linguistic analysis of SRSs written in natural language. RSL-IL provides several constructs that are logically arranged into viewpoints according to the specific RE concerns they address, and are organized according to two abstraction levels: business and system levels (Ferreira & Silva, 2013).

Despite sharing the same background and technologies, RSL-IL4Privacy was recently defined independently of the RSL-IL language and with the only purpose to support the rigorous specification of privacy policies with multi-representations. As suggested in Fig. 1, a RSL-IL4Privacy policy is represented as a set of privacy Statements and other related constructs such as Services, Recipients, Private Data and Enforcements (Caramujo and Silva,

2015). The Statement is the key concept of the privacy-aware profile. This element describes what rules or actions are specified in a privacy policy, therefore it is considered a privacy requirement. It is also noteworthy that one Statement may refer several services and several privacy data (Service and PrivateData elements respectively). Each Statement can be classified into five different categories, according to its purpose (Caramujo and Silva, 2015): Collection (which data is collected); Disclosure (which data is disclosed and to what parties); Retention (how long data will be stored); Usage (what is the purpose of having the data); and Informative (with just generic information). This approach has been supported by an Eclipse plugin, called "RSLingo4Privacy Studio" and available from its GitHub repository (https://github.com/RSLingo/RSLingo4Privacy).

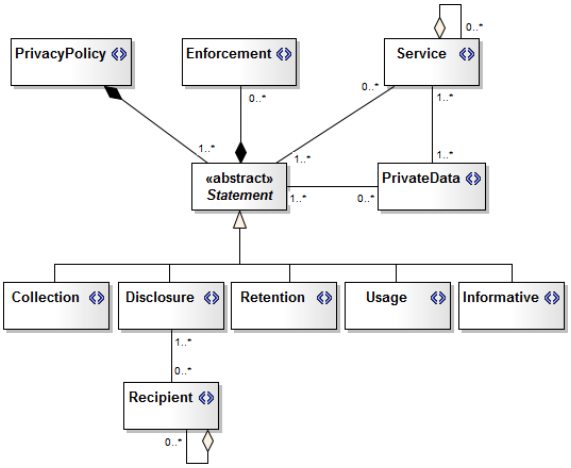


Figure 1: RSL-IL4Privacy metamodel (partial view).

2.2 Eddy Language

Eddy is a formal language for specifying privacy requirements (Breaux et al., 2014). Eddy is expressed based on Description Logics (DL) (Baader et al., 2003) that allows specifying actors, data, and data-use purpose hierarchies based on the DL subsumption. It also allows to specify the modality (i.e., permission and prohibition) of such data purposes and then automatically detects conflicts between what it is permitted and what it is prohibited. Eddy language is supported by the Eddy engine (on top of an OWL reasoner) available at https://github.com/cmu-relab/eddy.

3 RSLingo4Privacy APPROACH

A privacy policy (PP) is a technical document that states multiple privacy-related requirements that websites and mobile apps should show and respective organizations should satisfy. These requirements are usually defined as ad-hoc natural language statements, meaning that there is not a rigorous and consistent way to specify and validate them. In spite the advantages of natural language as a flexible, universal, and human proficiency at using it to communicate with each other, there are some well-known restrictions such as the difficulty to automatically analyse and validate the quality of those specifications.

RSLingo4Privacy approach supports the specification of privacy policies giving concrete guidance to improve their quality. RSLingo4Privacy includes several processes (supported by respective tools), namely:

P1: automatic text classification and extraction;

P2: visualization and authoring;

P3: analysis and quality validation; and

P4: (re)publishing.

RSLingo4Privacy is a multi-language approach that uses the following privacy-aware languages (as introduced in Section 2): RSL-IL4Privacy and Eddy. Fig. 2 overviews RSLingo4Privacy approach as a top-level BPMN business process diagram.

If a given (ad-hoc natural language) policy exists, the process P1 applies complex text classification and text extraction techniques to automatically produce the equivalent specification in RSL-IL4Privacy (P1 is further discussed in Section 4). In addition or otherwise, if that policy does not exist, the RSLingo4Privacy approach starts directly with process P2 to allow visualizing and authoring the policy in a rigorous and consistent way based on the RSL-IL4Privacy language (P2 is further discussed in Section 5). Process P3 takes as input both RSL-IL4Privacy and Eddy specifications, and provides analysis and validation features, producing, for example an analysis report with errors and warnings that can be taken into consideration during these authoring and validation processes.

Finally, when the quality of the policy specified in RSL-IL4Privacy is appropriated, the process P4 is responsible for producing an improved version of the policy, specified again in natural language but in a more consistent and high-quality manner. This publishing process is based on the Apache POI framework (https://poi.apache.org/).

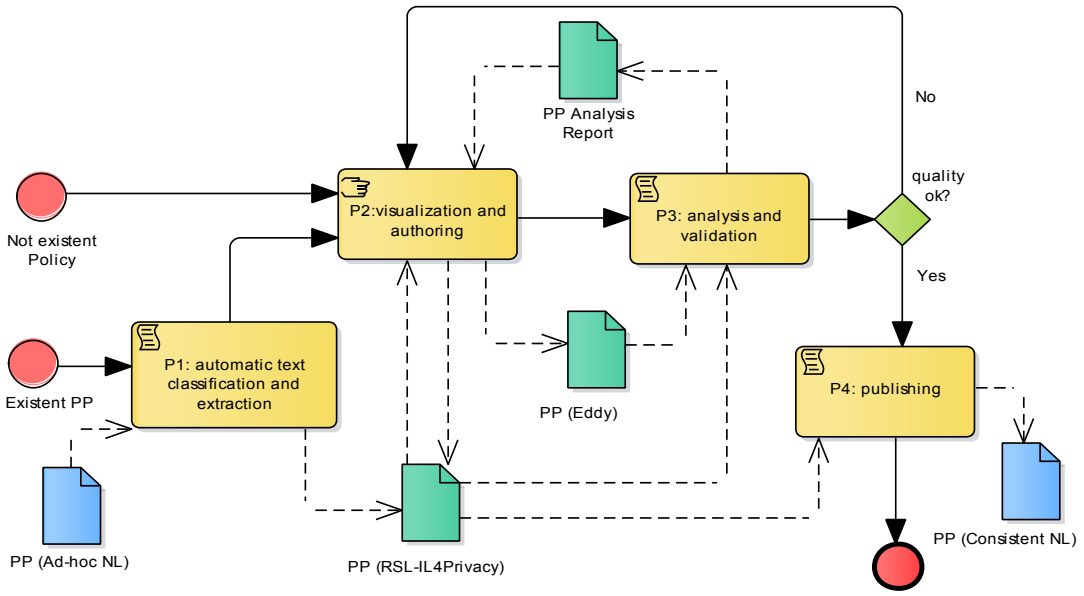


Figure 2: RSLingo4Privacy approach (defined with a BPMN business process diagram).

Due to space constraints this paper focuses the discussion in just the first two processes, i.e. P1 and P2. This approach has been evaluated against the policies of most popular systems; however, for the sake of briefness we just consider some statements taken from the Facebook privacy policy (https://www.facebook.com/policy) for supporting the presentation and discussion of current results in the following sections.

4 TEXT CLASSIFICATION AND EXTRACTION (P1)

One of the goals regarding the privacy policies of popular information systems is to govern users' personal information by describing a set of actions or rules for managing it in terms of how the company shares, keeps or uses such data. These policies are written using natural language and do not have any specific format attached, i.e., the number of sections and paragraphs, as well as the length or the type of language used, is quite contrasting, varying from one privacy policy to another. Being an exhaustive and very detailed document, privacy policies pose problems for end-users (e.g., poor understanding of the different personal data flows within a policy) but also for developers and service providers (e.g., difficulty in extracting the right requirements from a policy).

This process P1 intends to optimize the process of analysing privacy policies. First, through the automatic classification of the different statements that comprise a policy into a set of five distinct types. Second, by automatically extracting some relevant elements from those classified statements. Both the statement types and relevant elements are defined beforehand in RSL-IL4Privacy.

4.1 Automatic Text Classification

The task of classifying statements according to a given type is truly important under the scope of RSLingo4Privacy, since each kind of statement has different features and raises different concerns. However, doing it manually is very time-consuming and requires a lot of human-effort, which in itself lowers people's motivation, therefore increasing the probability of making mistakes during the analysis. Streamlining this process by having an automatic classification of the statements in a privacy policy while achieving reliable results is of the utmost importance.

4.1.1 The Classification Model

According to the RSL-IL4Privacy metamodel (see Fig. 1), statement sentences can belong to one of five categories: Collection, Disclosure, Retention, Usage, and Informative. Our goal is to build a classifier that, given a sentence from a specific policy, can determine to which of these it belongs. The classifier architecture is depicted in Fig. 3.

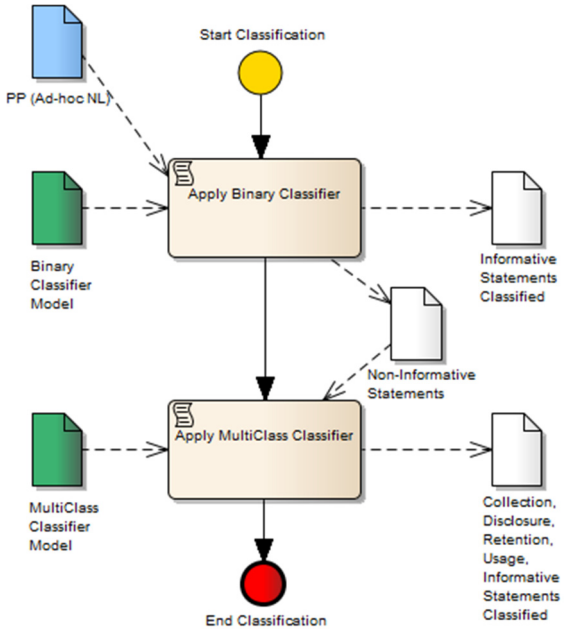


Figure 3: Statement classifier architecture.

The classifier contains two main components, each containing its own specialized classification model. The Binary Classifier Model is used to determine if a given sentence is of class Informative or not. Informative sentences usually contain very generic text and, thus, can hamper the determination of the remaining classes. For this reason, this first filtering step is taken. Once a sentence is classified as non-Informative, it is passed as input to the MultiClass Classifier Model, which determines its class among the remaining four categories. Even though the main goal of this second classifier is to label a non-informative statement as Collection, Disclosure, Retention and Usage, it also has the ability of determining if a non-informative is "informative". By doing this specific classification step two times, we get another opportunity to properly classify an informative statement that may have been labelled incorrectly as non-informative by the first classifier.

Each sentence is represented by its constituent words and their TF-IDF weights (Ramos, 2003), after some preprocessing. This preprocessing includes: discarding words with less than 3 characters, pruning words that occur in less than 3 documents and in more than 300 sentences, removal of stopwords, reduction to word stems, and generation of 2-grams (i.e. sequences of two consecutive words). After this preprocessing, the most informative words are selected using a function that assigns – for each word - the coefficients of a hyperplane calculated by a Support Vector Machine (Cortes and Vapnik, 1995) for the Binary classifier and Information Gain (Quinlan, 1986) for the MultiClass classifier. The best results for the Binary classifier were achieved with the 700 words with the highest values, whereas those for the MultiClass classifier were achieved with only 600 words.

4.1.2 Data

One of the biggest problems concerning the automatic classification of privacy policies is the lack of annotated privacy policies available for common use (Ammar et al., 2012). To carry out this experiment, we ourselves collected the statements (i.e., sentences) from 6 privacy policies of well-known websites: Facebook, LinkedIn, Zynga, Dropbox, IMDb and Twitter. We manually classified each statement according to their category and ended up with a dataset comprised of 598 examples. Table 1 summarizes the distribution of examples throughout the various categories.

Table 1: Number of statements per type.

Type	Nr. of Statements
Collection	78
Disclosure	114
Retention	64
Usage	92
Informative	250

4.1.3 Preliminary Results

The system with two classifiers have been tested to measure the solution's feasibility and some preliminary results are already available. All tests were performed using 5-fold cross-validation. The effectiveness of the proposed system was measured according to the standard metrics of accuracy, precision, recall and the F-score. Table 2 shows the system performance, per statement type, in terms of such evaluation metrics. All values are quite high, particularly those of precision, which illustrates the ability of the system to correctly discriminate between statement types. However, despite being subject to classification by both classifiers, the "informative" type of statements still have a lower precision in comparison with the remaining types.

The proposed solution returned an accuracy value of 84.28% which means that only less than 20% of the total number of statements are wrongly classified. On the other hand, the Binary classifier

Table 2: System performance, per statement type, in terms of precision, recall, and F-score. The last column shows the overall system accuracy.

Type	Prec.	Rec.	F-score	Acc.
Collection	84.28%	70.51%	76.78%	84.28%
Disclosure	90.28%	78.95%	84.41%	
Retention	92.85%	75.00%	82.98%	
Usage	94.38%	72.83%	82.22%	
Informative	67.91%	97.90%	90.19%	

on its own has a global accuracy of 82.61%, whereas the MultiClass classifier holds an overall accuracy of 70.73%.

4.2 Automatic Text Extraction

Knowing the type of a statement gives a better insight on the different actions that apply to the users' personal information. However, it is necessary to automatically extract other pieces of knowledge from a privacy policy, in order to get a more in-depth understanding of how the users' information is in fact handled and governed.

The disclosure of personal information is a sensitive topic, thus it is crucial to discover the various entities that end up receiving information that is shared by the service provider. In addition, it is also necessary to grasp which information concerning users is after all disclosed, collected and retained. Thus, our priority is to extract, from each sentence, the elements of RSL-IL4Privacy "Recipient" and "PrivateData". A methodology that allows one to automatically detect these kinds of data, which may not be clearly specified or grouped together in the policy, plays an important role on the process of analysing and validating a privacy policy in RSLingo4Privacy.

Discovery and extraction of such elements will be performed through Conditional Random Fields (CRF) (Lafferty, McCallum and Pereira, 2001). A CRF is a framework for building probabilistic models to segment and label sequence data, i.e., it intends to find a label Y that maximizes the probability P(Y|X) for a given sequence data X. Each attribute of X receives a value from a feature function that associates such attribute with a possible label. Each feature holds a weight that represents its strength for the proposed label (Ceri et al., 2013): positive values mean a good association between the function and the label, negative values mean otherwise, and a value of 0 means that the feature function does not have an influence on the label identification. In short, CRFs provide a powerful and flexible mechanism for exploiting arbitrary feature sets along with dependency in the labels of neighbouring words (Sarawagi, 2008). [This task of entity extraction is still in its initial implementation phase.]

5 VISUALIZATION AND AUTHORING (P2)

As mentioned above, RSL-IL4Privacy allows specifying policies in a rigorous way. However, to provide a good support to both technical and non-technical stakeholders, a visualization and authoring environment is required. Such tool should provide common features that already exist in popular and general-purpose text editors, but also features that are found in language-specific tools such as parsers, linkers, compilers or interpreters. Due to these reasons we decided to implement such environment on the top of the Xtext framework.

5.1 Domain-specific Authoring Tool

Xtext is an open-source framework for developing domain specific languages (DSLs) that covers all aspects of language implementation such as parsers, linkers, compilers, interpreters and full-blown IDE support based on Eclipse (Bettini, 2013; http://xtext.org).

In addition, Xtend code generator can be used with the Xtext DSL to generate code/text to other languages such as Eddy, XML, DOC, and so on. The task of writing the generator is greatly simplified by the fact that Xtext automatically integrates the generator into the Eclipse infrastructure. As soon as running the Xtext grammar, a code generator is created into the runtime project of the DSL, and Java Beans will be defined for each entity of the DSL's domain model (Bettini, 2013).

The rules of the grammar are defined to describe the key entities and their relations. Each Entity has a name and some properties. Fig. 4 shows the partial RSL-IL4Privacy grammar definition for Collection and Private Data. After defining the grammar, we need to execute the code generator that derives the various language components, generates the parser and some additional infrastructure code.

```
PrivateData:
   'PrivateData' name=ID '{'
   'Description' privatedata=STRING ','
   'Type' PrivateDataKind=('PersonalInformation'| 'UsageInformation')',' attribute+=Attribute* '};';
 Attribute:
    'Attribute' name=STRING 'Description' attributeName=STRING (',')?;

Collection:
    'Collection' name=ID '{'
    'Description' description=STRING ','
    'Condition' condition = STRING','
    ('PartOf Collection' partof=[Collection]',')?
    ('RefersTo PrivateData' refprivatedata+=RefPrivateData*',')?
    ('RefersTo Service' refertoservice+=ReferToService*',')?
    ('RefersTo Enforcement' refertoEnforcement+=RefertoEnforcement*',')?
    'Modality' modalitykind=('Permission'|'Obligation'|'Prohibition') '};';
```

Figure 4: Xtext Grammar of RSL-IL4Privacy (partial view).

Table 3: Matching keywords for RSL-IL4Privacy and Eddy grammars.

Language	Modality	Action	Datum	Source	Target	Purpose
RSL-IL4Privacy	Permitted, Forbidden	COLLECT, USE, TRANSFER, RETAIN	(RefersTo) PrivateData	-	(RefersTo) Recipient	(RefersTo) Service
Eddy	P, O, R	Collection, Usage, Disclosure, Retention	D	FROM	TO	FOR

5.2 Model-to-Model Transformation (RSL-IL4Privacy to Eddy)

A RSL-IL4Privacy to Eddy generator was defined in the context of the Xtext framework. With this feature it is possible to generate Eddy specifications from equivalent RSL-IL4Privacy specifications.

To define this generator we had to find all the matching concepts between both RSL-IL4Privacy and Eddy grammars.

As discussed, a privacy policy specified using RSL-IL4Privacy encompasses a set of privacy elements: “Statement”, “Service”, “Recipient”, “PrivateData” and “Enforcement”. The single definition of a statement (i.e., its description, modality – forbidden or permitted) encloses the various associations with the remaining elements that are, in their turn, defined on the bottom of the privacy policy in RSL-IL4Privacy. A privacy policy in Eddy, on the other hand, is represented with a specification header (“SPEC HEADER”) and the following specification body (“SPEC POLICY”). The header aggregates the prior definitions of three elements: “P” for Purpose, “A” for Actor and “D” for Datum. The statements are then described on the body. Each statement has a modality (“P” indicates permission, “O” indicates obligation and “R” indicates prohibition), the action verb, the Datum, the source (“FROM”), the target (“TO”) and the Purpose (“FOR”). Based on the description of the different elements and keywords from both languages, it is possible to map the following concepts: the “PrivateData” can be considered as Datum, the “Service” as Purpose and the “Recipient” as Actor (target). Since the source (“FROM”) refers to the service provider, there is not a direct match between concepts in the two languages. Some relations between both grammars are clarified in Table 3.

The RSL-IL4Privacy to Eddy converter is defined on the top of the Xtend code generator framework. So, Eddy specifications are automatically created in Eclipse Editor based on equivalent RSL-IL4Privacy specification.

5.3 Simple Example based on the Facebook Policy

The following shows two Facebook’s statements represented in both Ad-hoc NL, RSL-IL4Privacy and Eddy languages. The ad-hoc natural language statements are shown in Fig. 5 and Fig. 6. The type of statement st1 is Collection that specifies what personal information will be collected by the service provider and st19 is a statement of type Disclosure that explicitly defines which information is shared to other external entities or third-parties or, in this case, which information is not shared to those entities.

The action using phrase heuristics (verbs) indicates which action should be assigned (e.g., “collect” indicates a COLLECT action and “share” indicates a TRANSFER action). The modal keywords “will” and “will not” infer the modality of permission and prohibition, respectively. Besides, the datum, purpose and target are clarified on these statements.

The definition of the mentioned statements in Eddy and RSL-IL4Privacy specifications are shown respectively in Fig. 7 and Fig. 8.

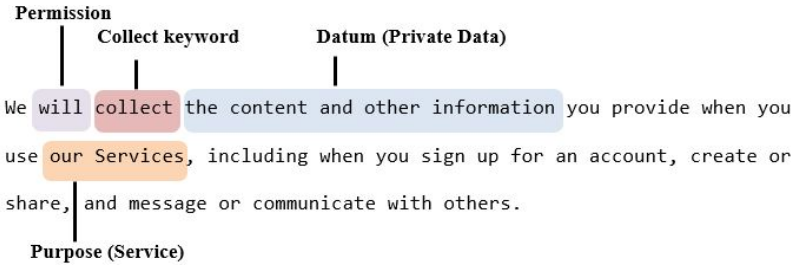


Figure 5: Statement st1 of Facebook Policy.

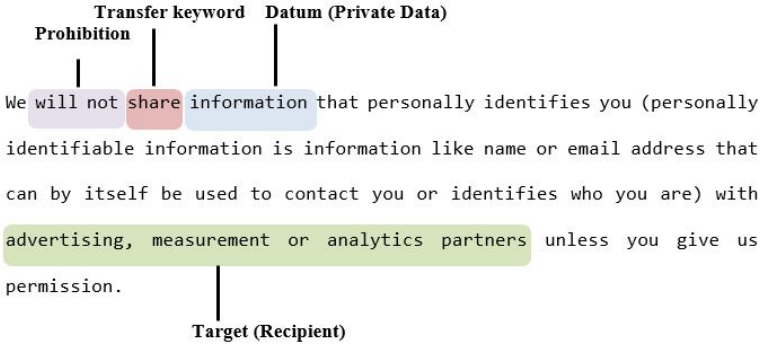


Figure 6: Statement st19 of Facebook Policy.

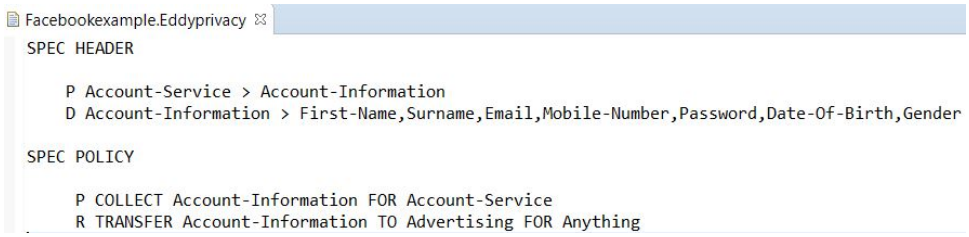


Figure 7: Eddy representation for Facebook's statement St1 and St19.

Table 4: Comparison of privacy-aware specification languages.

Language	**Domain**	**Abstract Syntax**, defined as a…	**Concrete Syntax,** represented by…	**Semantics**
RSL-IL	Generic	Grammar	Textual	Declarative
RSL-IL4Privacy	Data Privacy	UML Profile + Grammar	Graphic + Textual	Declarative
Eddy	Data Privacy	Grammar	Textual	OWL-DL
P3P/APPEL	Web Privacy	XML schema	Textual	Declarative
KAoS	Generic	DAML (XML schema)	Textual	OWL
Rei	Generic	Prolog* constructs	Textual	OWL

Table 5: Comparison of privacy-aware specification approaches.

Approach	**Languages**	**Tool Support**			
		Text Extraction	**Visualization & Authoring**	**Analysis & Validation**	**Publishing**
RSLingo4 Privacy	RSL-IL4Privacy + Eddy	Yes	Yes (Eclipse xText-based)	Yes (intra and inter policies)	Yes
Eddy	Eddy	No	Yes (General purpose text editor)	Yes (intra and inter policies)	No
P3P/APPEL	P3P/APPEL	No	Yes (General purpose text editor)	Yes (inter policies)	No
KAoS	KAoS	No	Yes (KPAT)	Yes (inter policies)	No
Rei	Rei	No	Yes (General purpose text editor)	Yes (inter policies)	No

Facebookexample.rslil4privacy

```
PrivacyPolicy "Facebook" Last revised: 30 Jan 2015;

Collection st1 {
     Description "We collect the content and other information you provide when you use our Services,
     including when you sign up for an account, create or share, and message or communicate with others.",
     Condition "Users must accept Statement of Rights and Responsibilities (including Data Policy)",
     RefersTo PrivateData PD1,
     RefersTo Service S1,
     Modality Permission };
Disclosure st19 {
     Description "We do not share information that personally identifies you (personally identifiable
     information is information like name or email address that can by itself be used to contact you
     or identifies who you are) with advertising, measurement or analytics partners unless you give us
     permission.",
     Condition "Users must accept Statement of Rights and Responsibilities (including Data Policy)+
     grant extra permission",
     RefersTo Recipient-Target R3,
     RefersTo PrivateData PD1,
     Modality Prohibition };

PrivateData PD1 {
     Description "Account-Information",
     Type PersonalInformation,
     Attribute "First-Name" Description "first name",
     Attribute "Surname" Description "surname",
     Attribute "Email" Description "email",
     Attribute "Mobile-Number" Description "mobile number",
     Attribute "Password" Description "password",
     Attribute "Date-Of-Birth" Description "date of birth",
     Attribute "Gender" Description "gender"};

Recipient R3 {
     Name "Advertising",
     Description "Advertising, measurement or analytics partners",
     Scope External,
     Type Organization };

Service S1 {
     Description "Account-Service",
     RefersTo PrivateData PD1,};

Enforcement En1 {
     Name "Activity-Log-tool",
     Description "Users are able to manage the content and information shared when using Facebook",
     Type Tool };
```

Figure 8: RSL-IL4Privacy representation for Facebook's statement St1 and St19.

6 RELATED WORK

Other approaches and privacy-aware languages for specifying privacy policies can be considered in an analysis of related work, namely P3P/APPEL, KAoS, and Rei. Table 4 gives a brief comparison of these languages, also with RSL-IL4Privacy and Eddy included in the context of the RSLingo4Privacy approach. Furthermore, Table 5 provides a comparison of the more high-level perspective concerning the process of privacy policies specification when using the aforementioned languages.

6.1 P3p/Appel

The Platform for Privacy Preferences, P3P, is an XML-based language that allows websites to express their privacy practices in a standard format (http://www.w3.org/TR/P3P). This format intends to provide user agents with the ability to easily access and interpret such practices, hence encoding them in a machine-readable format. APPEL (http://www.w3.org/TR/P3P-preferences) complements P3P by specifying a language that describes collections of preferences regarding P3P policies between P3P agents. P3P gives an exhaustive characterization of a policy by defining a set of elements about such policy. However, the lack of a well-defined semantics for P3P lead to an unclear separation between the elements described in a P3P policy and vague definition of what data is collected and retained, and which part of that data is disclosed to external entities.

6.2 KAoS

KAoS is a collection of componentized services compatible with popular agent frameworks (Uszok et al., 2003). KAoS policy services play a very important role because they deal with the whole policy life cycle by allowing the specification,

management, handling of conflicts, and enforcement of such policies within multiple domains. KAoS uses Web Ontology Language (OWL) as a central policy ontology, which allows the definition of the main policy-related concepts but also provides application developers with the possibility of extending and adding application-specific concepts (i.e., specific vocabulary) that may be useful when defining particular policies (e.g., privacy policies). Conflict detection occurs at specification time and relies on algorithms that are embedded into KAoS (Tonti, 2013).

6.3 Rei

The Rei policy language is a logic-based language, modelled on deontic concepts of rights, prohibitions, obligations and dispensations (Kagal et al., 2003). Rei is not tied to any particular application and supports the addition of domain-specific information, hence allowing the specification of different kinds of policies (including privacy policies). The Rei framework provides means to reason about policy specifications but it does not provide an enforcement model (Tonti, 2013). Even though it can detect conflicts, Rei does not have the proper tools for enforcing policies by preventing some entities (i.e., subjects) from performing unauthorized actions, for instance.

Most of the languages discussed in this section were developed with the goal of having a privacy policy written in a machine-readable format that allow one to reason about such policies. However, if we consider such languages within a privacy requirements specification approach, they do not encompass the common case where privacy policies are already written using natural language and the fundamental idea is to come up with an approach that deals with the whole process: get an existing privacy policy, process and extract the desired information and apply the new knowledge producing better versions of the current privacy policy. On the other hand, due to their syntax and semantics, they have no advantages to the final end-users of the systems (with regard to their own understanding of the policy itself) and developers need specific assistance for policy specification and interpretation (Tonti, 2013). For these reasons, these privacy-aware specification languages, although providing mechanisms to analyse and validate policies, lack the flexibility for being used in a more broad approach which contemplates the specification of privacy policies.

7 CONCLUSIONS

This paper proposes and discusses the RSLingo4Privacy approach that intends to improve the specification and analysis of privacy policies. RSLingo4Privacy complements the current state-of-the-art by providing a clear and plain approach for the specification of such requirements with multiple representations while taking into account the importance of having requirements documented in a format as close to natural language as possible. The validation with some case studies showed so far the adequacy of this approach (including its RSL-IL4Privacy and Eddy formal languages and respective tools) for the purpose discussed in the paper. The different representations, for distinct levels of formality, express the flexibility and reliability which is desired for these languages.

RSLingo4Privacy approach includes four key processes with respective tool support. Of these processes only two are discussed in the paper, namely: (P1) the automatic classification of statements and extraction of text snippets from original policies into equivalent specifications, and (P2) the visualization and authoring of these requirements in a consistent and rigorous way based on the RSL-IL4Privacy intermediate language.

Process P1 includes two tasks in sequence. The first task automatically classifies a set of statements into a set of five distinct categories. The second task automatically extracts the relevant elements from the original statements into equivalent RSL-IL4Privacy statements.

On the other hand, Process P2 includes several tasks, mainly related the visualization, authoring, but also syntactic analysis and validation of RSL-IL4Privacy policies. This process is supported by a domain-specific text editor that implements the RSL-IL4Privacy language on the top of the Xtext framework. Consequently, this tool provides relevant features to both technical and non-technical stakeholders in their collaborative work in what concerns the definition, understanding, analysis and (re)publishing of these policies.

The other two processes, i.e. P3 and P4, will be discussed in future publications. In addition, the main public results of this project are available at RSLingo4Privacy's GitHub repository (https://github.com/RSLingo/RSLingo4Privacy).

Several issues may be considered for future work such as the following. First, more extensive experiments should be achieved to better evaluate the effectiveness of the process P1, particularly in what concerns the automatic text extraction task. Second,

we should research techniques to manually and then automatically evaluate the quality of these privacy policies. For example, how can we evaluate the quality of a specific policy. Further research and guidelines may help companies to properly specify these policies. Third, and consequence from the second issue, we should include the ability to analyze not just one but a set of inter-related policies and automatically identify inconsistencies among the requirements stated in these policies, that increasingly appear in multi-tier systems, in which each tier may be owned and operated by a different party, and raising additional problems such as over-collection and repurposing (Breaux et al., 2015).

ACKNOWLEDGEMENTS

This work was partially supported by national funds under FCT projects UID/CEC/50021/2013, EXCL/EEI-ESS/0257/2012, CMUP-EPB/TIC/0053/2013 and the project TT-MDD-Mindbury/2014.

REFERENCES

Ammar, W., et al., 2012. Automatic categorization of privacy policies: A pilot study. In *School of Computer Science, Language Technology Institute, Technical Report CMU-LTI-12-019*.

Baader, F., Calvenese, D., McGuiness, D. (eds), 2003. *The description logic handbook: theory, implementation and applications*. Cambridge University Press.

Bettini, L., 2013. *Implementing Domain-Specific Languages with Xtext and Xtend*. Packt Publishing Ltd.

Bird, S., Klein, E., Loper, E., 2009. *Natural Language Processing with Python*. O'Reilly Media, 1st edition.

Breaux, T.D., Baumer, D.L., 2011. Legally 'Reasonable' Security Requirements: A 10-year FTC Retrospective. *Computers & Security*, 30(4):178-193.

Breaux, T. D., Hibshi, H. and Rao, A., 2014. Eddy, a formal language for specifying and analyzing data flow specifications for conflicting privacy requirements. *Requirements Engineering*, 19(3):1–27.

Breaux, T. D., Smullen, D., Hibshi, H., 2015. Detecting Repurposing and Over-collection in Multi-Party Privacy Requirements Specifications. In *Proceedings of IEEE International Requirements Engineering Conference (RE'15)*.

Caramujo, J., Silva, A. R., 2015. Analyzing Privacy Policies based on a Privacy-Aware Profile: the Facebook and LinkedIn case studies. In *Proceedings of IEEE CBI'2015*, IEEE.

Ceri, S. et al., 1995. *Web Information Retrieval*. Springer, 2013.

Cortes, C. and Vapnik, V., 1995. Support-vector networks. *Machine Learning*, 20(3):273-297.

Cunningham, H., 2006. Information Extraction, Automatic. In *Encyclopedia of Language & Linguistics*, volume 5. Elsevier, 2nd edition.

Davis, A. M., 2005. *Just Enough Requirements Management: Where Software Development Meets Marketing*. Dorset House Publishing, 1st edition.

Emam, K., Koru, A., 2008. A Replicated Survey of IT Software Project Failures. *IEEE Software,* 25(5):84-90.

Farrell, C.B., 2011. FTC charges deceptive privacy practices in Google's rollout of its buzz social network. In *U.S. Federal Trade Commission News Release*, March 30.

Ferreira, D., Silva, A. R., 2012. RSLingo: An Information Extraction Approach toward Formal Requirements Specifications. In *Proc. of the 2nd Int. Workshop on Model-Driven Requirements Engineering*, IEEE CS.

Ferreira, D., Silva, A. R., 2013. RSL-IL: An Interlingua for Formally Documenting Requirements. In *Proc. of the of Third IEEE International Workshop on Model-Driven Requirements Engineering*, IEEE CS.

Ferreira, D., Silva, A. R., 2013a. RSL-PL: A Linguistic Pattern Language for Documenting Software Requirements. In *Proc. of Third International Workshop on Requirements Patterns*, IEEE CS.

Kagal, L., Finin, T. and Joshi, A., 2003. A policy language for a pervasive computing environment. In *Proc. of the 4th IEEE International Workshop on Policies for Distributed Systems and Networks*, 63–74.

Kovitz, B., 1998. *Practical Software Requirements: Manual of Content and Style*. Manning.

Lafferty, J., McCallum, A. and Pereira, F., 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the 18th International Conference on Machine Learning*.

Pohl, K., 2010. *Requirements Engineering: Fundamentals, Principles, and Techniques*, Springer.

Quinlan, J., 1986. Induction of Decision Trees, *Machine Learning*, 1(1):81-106.

Ramos, J., 2003. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*.

Robertson, S., Robertson, J., 2006. Mastering the Requirements Process, 2nd edition. Addison-Wesley.

Sarawagi, S., 2008. Information Extraction. *Foundations and Trends in Databases* 1(3):261-377.

Silva, A. R., 2014. SpecQua: Towards a Framework for Requirements Specifications with Increased Quality. In *Enterprise Information Systems*. Springer.

Silva, A.R., 2015. Model-Driven Engineering: A Survey Supported by a Unified Conceptual Model. *Computer Languages, Systems & Structures*, 43. Elsevier.

Sommerville, I., Sawyer, P., 1997. *Requirements Engineering: A Good Practice Guide*. Wiley.

Steel, E., Fowler, G. A., 2010. Facebook in privacy breach. *Wall Street Journal*, Oct. 18.

Tonti, G. et al., 2003. Semantic Web languages for policy representation and reasoning: A comparison of KAoS,

Rei, and Ponder. *The Semantic Web – ISWC*, 2870, 419–437.

Uszok, A. et al., 2003. KAoS policy and domain services: Toward a description-logic approach to policy representation, deconfliction, and enforcement. In *Proceedings of the 4th IEEE International Workshop on Policies for Distributed Systems and Networks*.

A Naked Objects based Framework for Developing Android Business Applications

Fabiano Freitas and Paulo Henrique M. Maia
Academic Master in Computer Science, State University of Ceara, Fortaleza, Brazil
fabiano.gadelha.freitas@gmail.com, pauloh.maia@uece.br

Keywords: Naked Objects, Framework, Android Application.

Abstract: Naked Objects is an architectural pattern in which the business objects are handled directly in the user interface. In this pattern, developers are responsible only for the creation of business classes and do not need to concern with the implementation of the other layers. Although used in several frameworks for making the development of web and desktop systems faster, there is still a lack of tools that add the benefits of that pattern to the creation of Android applications. This paper introduces JustBusiness, a Naked Objects based framework that aims at supporting the creation of Android applications through automatic generation of user interfaces and persistence code from mapping the business classes. Two case studies that describe how the framework was used and its evaluation are also provided.

1 INTRODUCTION

The attention that the Android platform has gained over the recent years has resulted in an increasing demand for applications, a situation justified by the growth of the Android community. According to data from Net Marketshare [1] for March 2015, Android is the most widely used platform on the planet, with a percentage of 47.51%, reaching more than a billion mobile devices among smartphones and tablets.

To answer the market demands, developers and companies are increasingly using frameworks and libraries that can ease the development of applications, thus helping to boost productivity. Among these, we can cite Ormlite (ICE, 2015c), a framework to simplify the access and communication between the Android application and the SQLite database, and Roboletric (ICE, 2010c), a framework to facilitate testing Android applications.

Although useful, those frameworks do not solve or mitigate one of the main application development bottlenecks, which is the creation of user interfaces (UIs) and CRUD (create, read, update and delete) code for business objects, since they do not provide automatic generation mechanisms of those artifacts. That task is already commonly performed to web (Milosavljević et al., 2003) and desktop systems (da Cruz and Faria, 2010). According to Pawson (Pawson, 2004), the development of user interfaces is, most of the times, the task responsible for a significant proportion of all the effort involved in developing an interactive business system due to not only the complexity of coding, but also to the time spent with the presentation details.

As a result, the developer must manually build each of those interfaces and CRUD code, which is a tiring, time consuming and error prone task. Moreover, if the user wants to migrate an existing application from another platform, he/she may have difficult on reusing the business classes code, which implies in a rework to create them on the Android platform.

To fill that gap, this paper presents JustBusiness[2], a framework for developing Android business applications that provides automatic generation of user interfaces and CRUD code from information obtained from the object-user interface mapping of the business classes. For that, the framework implements the architectural pattern Naked Objects (Pawson and Matthews, 2001) (Pawson, 2004), in which the main application parts (the domain objects) are displayed in the interface and the user can manipulate them directly by using those objects' methods. With JustBusiness, the developer is only responsible for implementing the business classes, while the framework performs the heavy task of generation and configuration of all necessary classes and files for running an Android

[1] https://www.netmarketshare.com

[2] Download available at https://jbframework.wordpress.com

Freitas, F. and Maia, P.
A Naked Objects based Framework for Developing Android Business Applications.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 348-358
ISBN: 978-989-758-187-8

application.

The rest of this paper is divided as follow: section 2 explains the main concepts of Naked Objects, while the related work are discussed in Section 3. Section 4 presents the JustBusiness framework, describing its architecture and key features. In Section 5, two case studies describing the implementation of the framework and its evaluation are shown. Finally, Section 6 presents the conclusions and future work.

2 NAKED OBJECTS

The Naked Objects pattern is an object-oriented approach where the domain objects are exposed in the user interface, and the user has the power to manipulate them directly by performing invocations of methods implemented by those objects (Pawson, 2004). In that approach, the code of all classes must respect the object behavioral completeness, one of the most important principles of the object-oriented paradigm and that states that objects must fully implement what they represent (Pawson and Matthews, 2002) (Pawson, 2004) (Raja and Lakshmanan, 2010). The application of that pattern removes from the programmer the need for implementing user interface or security and persistence mechanisms, making him/her responsible only for creating the application domain objects, which must implement all behavior they propose to represent completely.

According to Raja and Lakshmanan (Raja and Lakshmanan, 2010), the Naked Objects approach has three principles that characterize the pattern: (i) the whole business logic should be encapsulated by business objects; (ii) the user interface should reflect the business objects; and (iii) the user interface generation should be automated from the domain objects. Also according to the authors, those principles boost the software development cycle, ease the requirements analysis, bring greater agility and produce more efficient user interfaces.

Naked Objects is an alternative to 4-layers (presentation, control, domain and persistence) architectural pattern. It uses a ratio 1:1 between elements of different layers, where for each domain object there exists only one match in the other layers, while in the 4-layers pattern there may be more complex mappings between the layers (Brandao et al., 2012). The comparison between the two architectural patterns is illustrated in Figure 1.

In (Pawson and Wade, 2003), Pawson and Wade conducted a study about using Naked Objects in an agile software development process. Among the found benefits, the authors point out that the approach foster

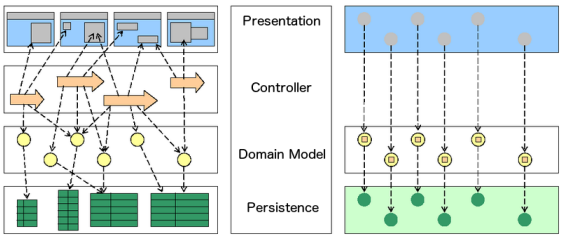


Figure 1: 4-layers (left) and Naked Objects (right) architectural patterns (Pawson, 2004).

the concept of exploration phase, in which users and/or customers, along with the development team, perform the UI prototyping simultaneously to the business objects modeling activity.

3 RELATED WORK

There are several frameworks and tools that promote the development of Naked Objects-based systems. We can highlight the following ones.

1. Naked Objects Framework (ICE, 2015a): one of the pioneer tools. It was developed in Java and used the concept of reflection, which was considered one of the main factors that influenced the choice of the programming language, according to its creator. It is focused on assisting the web and desktop application development.
2. Naked Objects MVC (ICE, 2015b): built upon the Microsoft ASP.NET platform, it aims at creating web applications, providing full-generation of user interfaces using ASP.NET MVC.
3. Apache Isis (ICE, 2010a): an open source framework implemented and focused on Java platform for rapid development of web applications. The Java reflection feature is used along with annotations, with which the programmer makes configuration specifications in the domain objects that will be treated by the framework. With this, Isis is responsible for managing the user interface, security and data persistence resources.
4. JMatter (ICE, 2008): a Java open source project that implements the Naked Objects pattern using, among other resources, Swing and Hibernate. It aims at helping the development of Java web applications by implementing their necessary infrastructure, including all CRUD structures, persistence and search mechanisms. As well as other similar frameworks, JMatter supports the UI automatic generation at runtime.

5. Entities (ICE, 2013a): framework implemented in Java to help developing web applications. One of its main benefits is the possibility of generation of customizable interfaces to the application through an object-user interface mapping. The generated interfaces can be customized through the annotations in the domain classes.
6. Isis Android Viewer (ICE, 2013b): this framework is not a tool to assist the development of Naked Objects-based applications, but rather an Android project to communicate to a web application developed with the Apache Isis framework. An Isis Android Viewer application creates representations of both user interface and persistence based on the domain objects present in the web application.
7. Naked Object for Android (ICE, 2010b): According to the tool's website, this is the first framework implemented to accelerate the Android application development using the Naked Objects principles. The tool, unlike the previous ones, does not support automatic generation of user interfaces nor persistence mechanisms. Currently it is not working, and there has not been recent updates nor maintenance activities.

An extension of the Naked Objects framework using annotations to allow the manipulation of higher-level abstractions, such as specialization of object relationships, is proposed by Broinizi *et al* in (Broinizi et al., 2008). According to the authors, using that approach to validate requirements brings as benefits the reduction of conceptual specification problems, like a weak identification of requirements, decreases the distance between domain and project experts, and allows the simultaneous exploration of conceptual data design and system requirements.

Keranen and Abrahamsson present in (Keranen and Abrahamsson, 2005) a study that compared two mobile development projects of the same mobile application for Java ME platform, where the first one used the traditional development, while the second one was developed using the Naked Objects Framework (NOF). As a result, there was a reduction of 79% in application code and 91% in interface code for the application with NOF. Despite those benefits, the authors concluded that NOF is still not mature enough to develop mobile applications.

Model-driven approaches for the creation and management of user interfaces, in which the software engineer develops the project of the system conceptual models from object oriented meta-models, are described in (Milosavljević et al., 2003) and (da Cruz and Faria, 2010). By applying pre-defined mapping rules, it performs the refinement and transformation of conceptual models, generating user interface and CRUD code from meta-models automatically.

All aforementioned approaches and tools do not support development for current mobile platforms, but rather only for web or desktop ones. Although the approaches proposed in (Keranen and Abrahamsson, 2005) and (Nilsson, 2009) use Naked Objects, they were designed for obsolete mobile platforms. Regarding the Android platform, the only found work was the Naked Objects for Android, but it has been discontinued. The framework proposed in this paper aims at bringing the benefits of Naked Objects to Android developers.

4 JustBusiness

JustBusiness is a framework based on the Naked Objects architectural pattern that provides support for the development of object-oriented business applications in the Android platform and that simplifies the migration of applications from other platforms, such as web ans desktop, to that mobile one. Like in other existing Naked Objects-based frameworks, such as the Naked Objects Framework, Apache Isis, JMatter and Entities, JustBusiness exposes the domain objects in the user interface, allowing users to manipulate them directly via invocations of methods implemented by those objects. Furthermore, it also removes from the programmer the responsibility of building user interfaces and persistence mechanisms, since it supports the automatic generation of those artifacts.

Besides being used for development and migration of business applications for the Android platform, JustBusiness can be used in the generation of the initial skeleton of the application. By providing automatic code generation, the developer can give up from using JustBusiness at any time in the development application process without the loss of source code that has already been produced.

More details about the JustBusiness framework are shown in the following sections.

4.1 Architecture

The framework has in its class structure, the abstract superclass *JBEntity*, which must be specialized by all business classes of the application project. The *JBEntity* class has no attributes and contains only two methods: *toPrimaryDescription* and *toSecondaryDescription*, which must be implemented by its subclasses, as shown in Figure 2.

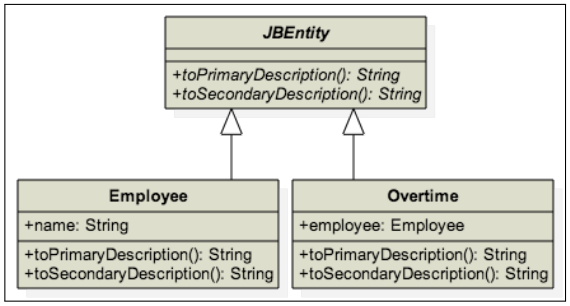


Figure 2: Extending the superclass JBEntity.

Those methods provide information about the object in the user interface, focusing on list screens (ListActivities). By default, the list of cells that are used by ListActivities has one or two fields to display information, as shown in Figure 3. Instead of using a single *toString* method, the structure with two methods supports more details in the interface, as well as the use of the two types of basic standard Android cells.

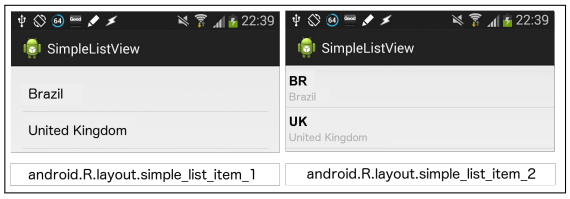


Figure 3: Listing cell standard Android platform.

The business classes should be implemented in a simplified way using a standard constructor, private attributes, and its respective *get* and *set* methods. The structure of the business classes is similar to a POJO (Plain Old Java Object) class, but differs from it because they extend the *JBEntity* superclass and use predefined notes in its construction.

JustBusiness uses a processor that analyzes, at compile time, each annotation used in the configuration of the business classes. Information obtained from the processing are stored in a data dictionary that contains all the information of classes, enumerations, attributes, methods and mapped parameters. When a "*clean and rebuild*" action is performed in the project, the saved information in the data dictionary are processed by code generators, which are responsible for creating and configuring all the needed files to deploy the application and run the project.

For each business class in the project, at least 25 files are created, as illustrated by Figure 4. Those files consist of control and data access classes and resource files, which are divided into layout and menu. For each of those files, there is a code generator that encodes them based on information obtained from the annotations. In addition, JustBusiness also creates a

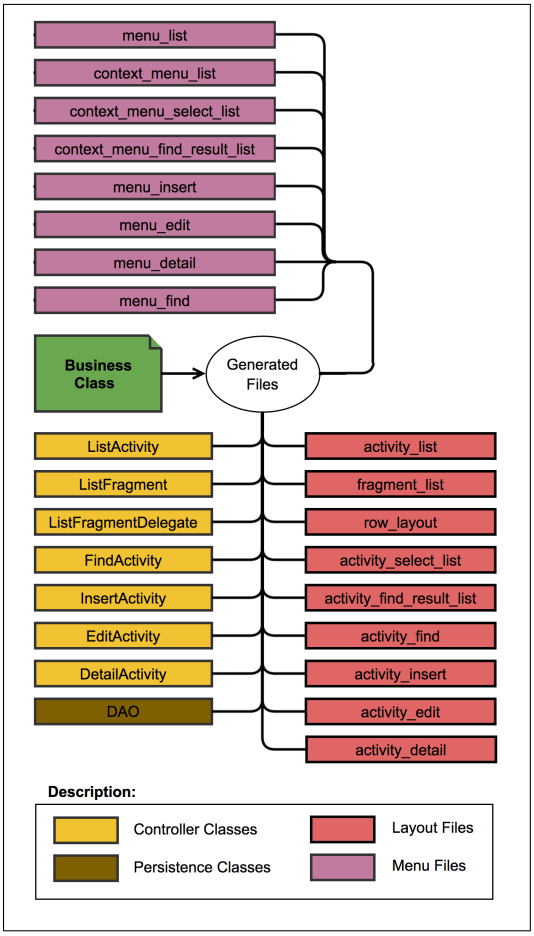


Figure 4: Structure of generated files to the business classes.

set of classes and resource files, based on information from annotations and business classes, for project organization and execution.

From the mapped information, three control classes are created to support the business application initial screen, which brings the list of entities contained in the application. The *dimension* and *styles* resource files are modified, bringing general information for dimensions and layout styles, respectively, while the *strings* file is modified to store all text content that will be used in the application. Furthermore, the *persistence* file, which contains persistence settings to access the database, is created. Finally, the *AndroidManifest* file is also altered.

For each new file creation or existing project file update, there is a code generator associated. The structure of project files generated or updated is shown in Figure 5.

4.2 Main Features

JustBusiness provides a set of annotations to enable

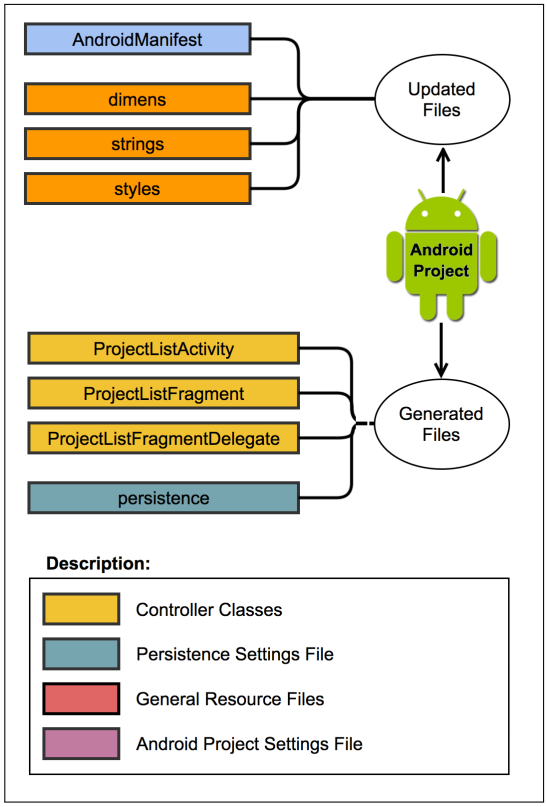


Figure 5: Structure of generated or changed project files.

the mapping information and to configure the business classes. Through those annotations, the programmer can detail information ranging from presentation settings in the UI to data persistence parameters.

In addition to the annotations feature, the framework uses a code generation mechanism that consists of a set of classes that, using information obtained from the mapping, generate automatically classes, resources and settings required for the production of an Android application.

4.2.1 Annotations

In order to configure and map information in the business classes, two sets of annotations have been defined: one for user interface and the other one for persistence. The former aims at providing details of the visual settings, while the latter details information for the object-relational mapping. The information and values passed through those annotations provide a greater level of detail about the mapped elements.

Annotations for User Interface. A single class, with its attributes and methods, does not have all necessary information for the complete generation of a graphical interface. To fill this gap, JustBusiness provides a set of annotations that allows the programmer to inform the missing information needed to perform that task.

Using the annotations, it is possible to define settings such as the business classes that will be recognized by the framework, the attributes that will be displayed on the screen and in which order, and the methods that will be available in the interface. JustBusiness uses the annotations *@Entity*, *@Attribute*, *@Action*, *@Parameter* and *@Enumeration* to identify and configure the classes, attributes, methods, parameters from methods and enumerations, respectively. Those annotations are detailed in Table 1.

Annotations for Persistence. The Android mobile platform, by default, uses the SQLite database, a very simple and limited database. One of its main limitations refers to the few supported data types (only INTEGER, TEXT, NONE, REAL and NUMERIC). Another important factor that should be considered is that the Android platform does not support the Java Persistence API (JPA) and does not recognize the *javax.persistence* package, which consists of a set of annotations and other classes aimed at mapping persistence information.

Given those limitations, JustBusiness has added some persistence features to provide object-relational mapping between business classes and SQLite database tables. For this, a set of annotations, inspired on the *javax.persistence* annotations, has been defined. Furthermore, the framework provides a mechanism for creating tables and SQLite database access from the information mapped with those persistence annotations. The set of annotations used for data persistence is listed in Table 2.

4.2.2 Automatic Code Generation

The framework supports the automatic generation of all necessary infrastructure for an Android application, including classes and user interface resources, and the SQLite database and data access classes. The programmer is responsible only for implementing the business classes and configuring them to use the framework. The code generation occurs at compile time, since, at that moment, interface classes and resource files are created, and some project configuration files, such as the *AndroidManifest*, are modified to incorporate the changes made by the framework.

Table 1: Annotations for User Interface.

Annotation	Description
@Entity	Describes and configures the classes that will be recognized by the JustBusiness. Parameters: **label** - Name that will be displayed for the entity **collectionLabel** - Plural name for the entity **icon** - Name of the image to be used as icon by the entity. The image must be in the folder (res/drawable)
@Attribute	Describes and configures the class attributes. Parameters: **name** - Name that will be displayed for the attribute **order** - Order in which the attribute is shown in the user interface **views** - Screens in which the attribute is displayed. It can be used more than one value. The possible values are: KindView.ALL: All screens KindView.INSERT: Entry screen KindView.EDIT: Update screen KindView.DETAIL: Detail screen KindView.FIND: Search screen
@Action	Describes and configures the class methods. Parameters: **name** - Name that will be displayed for the method **order** - Order in which the method is shown in menu
@Parameter	Describes and configures parameters of class methods. Parâmetros: **name** - Name that will be displayed for the parameter **order** - Order of the parameter in the form
@Enumeration	Describes enumerations.

Table 2: Annotations for Persistence.

Annotation	Description
@Table	Identifies a class and sets it as a table in the database.
@Id	Identifies an attribute as a key in the table.
@Column	Identifies and configures an attribute as a column in the table.
@JoinColumn	Identifies and configures a column as an attribute to perform a join operation with another table.
@Transient	Identifies an attribute that is not mapped in the table.
@OneToMany	Identifies and configures an attribute as a 1:N relationship.
@OneToOne	Identifies and configures an attribute as a 1:1 relationship.
@ManyToOne	Identifies and configures an attribute as a N:1 relationship.
@ManyToMany	Identifies and configures an attribute as a N:M relationship.
@JoinTable	Identifies and configures an attribute as a column for the operation join with another table.
@Enumerated	Identifies an attribute as an enumeration.
@Temporal	Sets an attribute that stores temporal information.

User Interface Code Generation. Through the information obtained from the mapping of classes, attributes, relationships, and methods using the aforementioned notes, the framework generates the whole user interface mechanism.

For each business class, the framework constructs the interfaces for inserting, editing, detailing and searching information as forms, where the components associated with each class attribute are arranged on the screen in a single vertical column according to the sequence determined by the developer in the business class. Those components can be enabled or disabled on the interface by setting the *@Attribute* annotation in the business class.

According to the Naked Objects pattern, changes are made exclusively in the business model, so the developer does not need to modify the interfaces directly. Therefore, in a project using JustBusiness, changes occur only in the business classes and, to incorporate that modifications in the project, the programmer only needs to recompile it, thus increasing the application's maintenance and evolution level.

Persistence Code Generation. By mapping the classes and their attributes and relationships using the proposed annotations, the framework generates the entire SQLite database automatically, including tables and keys. In addition, for each mapped class in the

project, the framework automatically generates a data access class using the Data Access Object (DAO) pattern.

Project Files Configuration. In addition to generating the control classes (Activities) and interface layout files, JustBusiness is also responsible for modifying the project configuration files, such as the *AndroidManifest*, which should be modified at compile time so that the project can identify all control classes that have been added, as well as receive information such as nomenclature and application icon. Besides the *AndroidManifest*, the resource files *strings*, *dimens* and *styles* are also modified.

Internationalization. The textual information mapped by the programmer using the annotations in business classes are compiled and stored in the *strings* resource file, which contains the words used within the application context. By using this approach, the application can be easily adapted to support a new language or dialect further.

4.3 Setting Up a Project with JustBusiness

To use the JustBusiness framework in an Android project, it is necessary to follow the steps depicted in Figure 6. As JustBusiness was designed, initially, to be used with Eclipse, we consider the use of that IDE to perform the following steps.

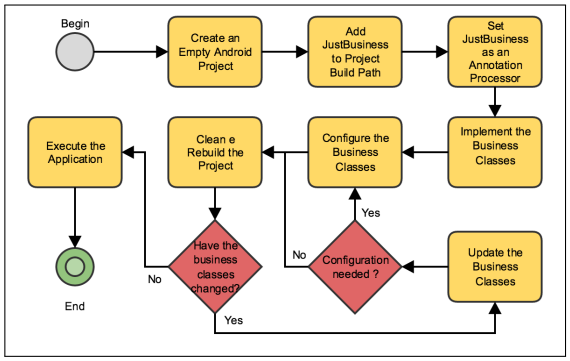


Figure 6: Flowchart for the use of JustBusiness.

Firstly, the developer must create an empty Android project and then configure it to recognize the framework. The project configuration consists of two activities: adding the framework to the project libraries and configuring JustBusiness as an annotation processor in the project properties.

The next step consists of implementing the business classes, which must inherit the *JBEntity* superclass and override its abstract methods. Subsequently, the business classes must be consistently annotated with annotations provided by the framework in order to build the interface and persistence mechanisms.

Finally, the project should be cleaned and rebuilt, since the code generation occurs at compile time. After that, the project is ready to be executed. If a new change in a business class is carried out, the developer must verify whether it is necessary to reconfigure the classes using the annotations and, if so, repeat the project clean and rebuild step.

5 CASE STUDY

To demonstrate the use of JustBusiness, a case study that consisted of the development of an application for request and approval of service overtime, as used in (Brandao et al., 2012), was carried out. The scenario starts with the employee requesting a service overtime, stating the justification and the initial and final date. The requests are firstly reviewed by the supervisor, who can either authorize or reject them. The authorized requests will be reviewed by the Human Resources (HR) sector to calculate and approve the payment of the hours. If HR has some questions, the request may return to the supervisor. Figure 7 is the resulting class diagram of the proposed case study.

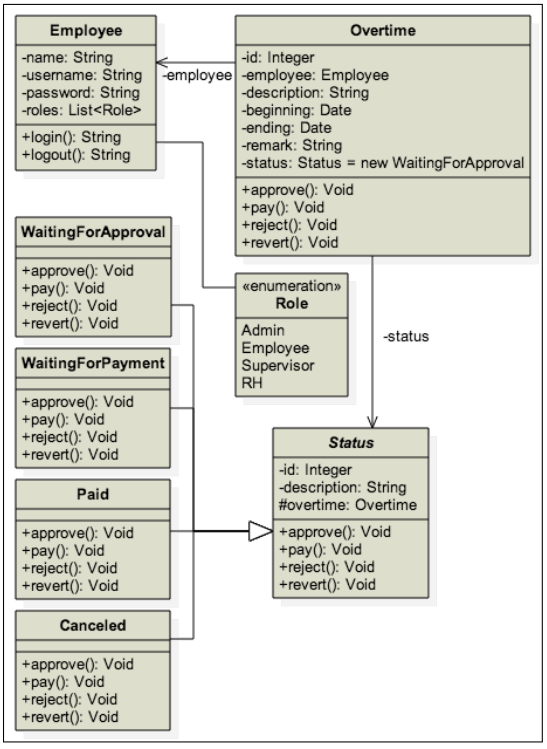


Figure 7: Class diagram. Source: (Brandao et al., 2012).

5.1 Applying JustBusiness to the Described Scenario

To use JustBusiness, the developer needs to implement only the business classes with their attributes, methods and relationships. He/she should firstly identify which business model elements are classes and which ones are enumerations, since there is a different treatment for each one. Classes must specialize the superclass *JBEntity*, have a default constructor with no arguments, *get* and *set* methods for each attribute, and must be mapped with the annotations @*Entity* and @*Table*. Enumerations should only be mapped with the annotations @*Enumeration* and @*Table*. Listing 1 shows part of the *Overtime* class source code. For the sake of simplicity, some details like *gets* and *sets*, as well as of some attributes and methods settings have been omitted.

In line 1 the @*Entity* annotation is used to indicate that *Overtime* class will be recognized by JustBusiness as a business class. Using it, the developer should inform the class *label*, that corresponds to the class name, and the *collectionLabel*, which is equivalent to the class plural name. In line 2, the @*Table* annotation is used to create the *overtime* table, whose name was informed in the *name* parameter, and to create the DAO classes with the SQL queries.

After, in line 4, the @*Id* annotation is used to indicate that the attribute *id* is a key in the *overtime* table. The @*Column* annotation determines, in line 5, that the id attribute correspond to the *id_overtime* column in the table. In line 6, the @*Attribute* annotation specifies that an element should be recognized as an attribute with label *Identifier*, has *order* 1, i.e., it will be first element on screen, and will appear just in the detail view in the user interface.

Then, in line 9, the @*ManyToOne* annotation specifies that an attribute represents a relationship N:1 with the *Employee* class, mapped through the *targetEntity* parameter. In the next line, the @*JoinColumn* annotation informs that an attribute will perform a join operation through the *id_employee* with the table mapped by the *Employee* class. In line 11, the @*Attribute* annotation specifies that an element should be recognized as an attribute with label *Employee*, has *order* 2, i.e., it will be second element on screen, and will appear in all views in the user interface.

Finally, the maps relating to methods *approve* and *pay* can be seen in the lines 24 and 29, respectively. The @*Action* annotation is used to identify a method that will be available for the user. The *name* parameter value is the title of the button that calls the method, while the *order* parameter specifies the position in the menu where the method appears.

```
1  @Entity(label="Overtime",collectionLabel="Overtimes")
2  @Table(name="overtime")
3  public class Overtime extends JBEntity {
4      @Id
5      @Column(name="id_overtime",nullable=false,unique=
         true)
6      @Attribute(name="Identifier",order=1,views={KindView
         .DETAIL})
7      Integer id;
8
9      @ManyToOne(targetEntity="business.Employee")
10     @JoinColumn(name="id_employee",nullable=false,unique
         =false)
11     @Attribute(name="Employee",order=2,views={KindView.
         ALL})
12     Employee employee;
13
14     //Annotations omitted
15     Date beginning;
16     Date ending;
17     String description;
18     String remark;
19     Status status = Status.WAITING_FOR_APPROVAL;
20
21     //Getters and Setters omitted
22     //toPrimaryDescription and toSecundaryDescription
         methods omitted
23
24     @Action(name="Approve",order=1)
25     public void approve() {
26         status.approve(this);
27     }
28
29     @Action(name="Pay",order=2)
30     public void pay() {
31         status.pay(this);
32     }
33
34     //Other methods omitted
35 }
```

Listing 1: Source Code of Overtime Class.

Listing 2 shows the simplified source code of the *Employee* class. Like the *Overtime* class, for simplicity, some details like *gets* and *sets*, as well as some attributes and methods settings have been omitted. To avoid repetition, only the annotations that have not been used in the the *Overtime* class will be explained.

In line 15, the @*ManyToMany* annotation specifies that an attribute represents a relationship N:N with the *Role* enumeration, mapped through the parameter *targetEntity*. In the next line, the @*JoinTable* annotation informs that the *roles* attribute will perform a join operation with the table mapped by the *Role* enumeration (shown in Listing 4) using the intermediate table *employee_role*.

```
1  @Entity(label="Employee",collectionLabel="Employees")
2  @Table(name="employee")
3  public class Employee extends JBEntity {
4      @Id
5      @Column(name="id_employee",nullable=false,unique=
         true)
6      @Attribute(name="Identifier",order=1,views={KindView
         .DETAIL})
7      Integer id;
8
9      //Annotationsomitted
10     String name;
11     String username;
12     String password;
13     List<Overtime> overtimes;
14
15     @ManyToMany(targetEntity="business.Role")
16     @JoinTable(name="employee_role",
17         joinColumns={@JoinColumn(name="id_employee",
18         referencedColumnName="id_employee")},
19         inverseJoinColumns={@JoinColumn(name="id",
20         referencedColumnName="id_role")})
```

```
21      @Attribute(name="Roles",order=5,views={KindView.ALL
          })
22      List<Role> roles;
23
24      //Getters and Setters omitted
25      //toPrimaryDescription and toSecundaryDescription
          methods omitted
26
27      @Action(name="Login",order=1)
28      public String login() {
29          return null;
30      }
31
32      @Action(name="Logout",order=2)
33      public String logout() {
34          return null;
35      }
36  }
```

Listing 2: Source Code of Employee Class.

As previously mentioned, the enumerations are identified and configured differently of the classes since they consist of limited entities with a structure already set. Listings 3 and 4 show the definition of the enumerations *Status* and *Role*, respectively. In both source codes, the *@Enumeration* annotation, in line 1, is used to identify the enumerations that will be recognized by JustBusiness, while the *@Table* annotation is used in line 2 to create a table in the database whose name will be the value of the *name* parameter.

```
1   @Enumeration
2   @Table(name="status")
3   public enum Status {
4       WAITING_FOR_APPROVAL{
5           public void approve(Overtime overtime) {
6               overtime.setStatus(WAITING_FOR_PAYMENT);
7           }
8           public void pay(Overtime overtime) {
9               //Donothing
10          }
11          public void reject(Overtime overtime) {
12              overtime.setStatus(CANCELED);
13          }
14          public void revert(Overtime overtime) {
15              //Donothing
16          }
17      },
18      WAITING_FOR_PAYMENT {},
19      PAID{},
20      CANCELED{};
21
22      public abstract void approve(Overtime overtime);
23      public abstract void pay(Overtime overtime);
24      public abstract void reject(Overtime overtime);
25      public abstract void revert(Overtime overtime);
26
27      @Override
28      public String toString(){
29          return this.name();
30      }
31  }
```

Listing 3: Source Code of Status Enumeration.

```
1   @Enumeration
2   @Table(name="role")
3   public enum Role {
4       ADMIN,
5       EMPLOYEE,
6       SUPERVISOR,
7       RH
8   }
```

Listing 4: Source Code of Role Enumeration.

Figure 8 displays the user interfaces for inserting and detailing the automatically generated information for the *Overtime* class from the information mapped in the class code, as shown in Listing 1. Figure 9 shows the user interfaces for the search operation and search result listing by type *Overtime* objects. Figure 10 displays the screen that contains the list of all objects of type *Overtime*. From that screen, the user can access the individual objects and perform operations on them.

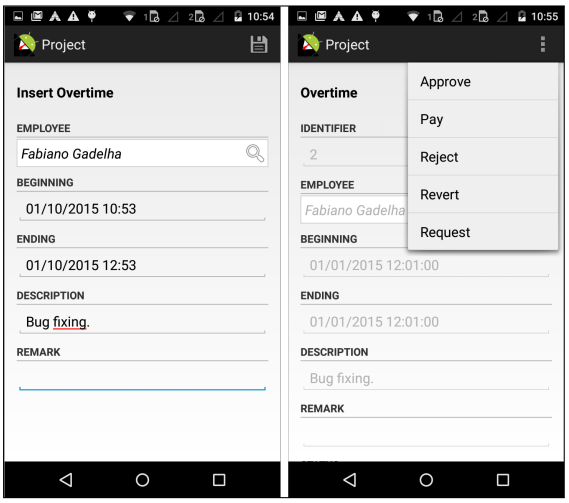


Figure 8: Entry and detail screens of object from type Overtime.

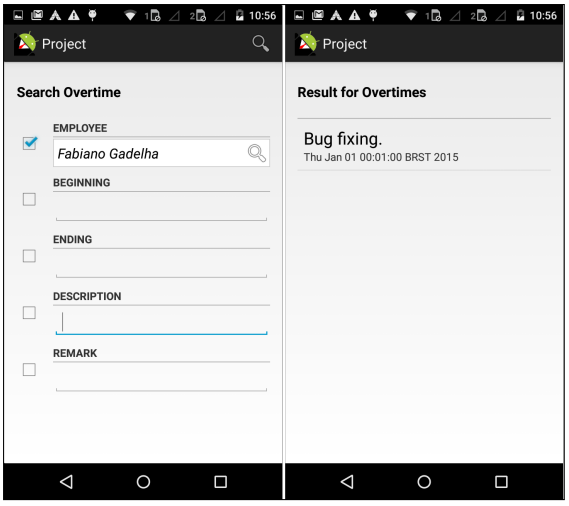


Figure 9: Search and result list screens of objects from type Overtime.

5.2 Evaluation

To assess the benefits of using JustBusiness framework, two comparative experiments have been conducted. For the first analysis, we developed two projects of the same Android application based on the scenario described in the previous detailed case study: the first one using the traditional development model, i.e., without

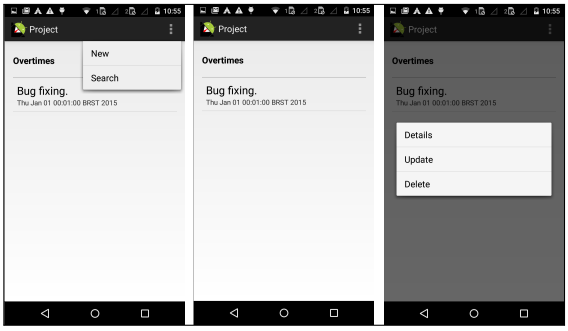


Figure 10: Listing screen of objects from type Overtime

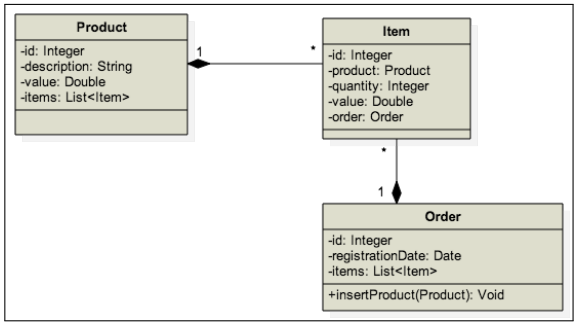


Figure 11: Class Diagram of the Second Case Study.

JustBusiness framework, while the second one using the framework. This assessment included a single programmer, who developed the two projects: initially the one without the framework and, subsequently, the other one using JustBusiness.

At the end of the implementation of the two projects, it was found that the project developed with JustBusiness achieved a reduction of approximately 96% of development time, 91% of written lines of code and 94% of files created by the developer. Comparative data for the two developed projects are detailed in Table 3.

Table 3: Comparative analysis of JustBusiness and the Traditional Development.

Type of Development	JustBusiness	Traditional
Time	27 minutes	720 minutes
Lines written by Developer	376	4315
Files created by Developer	4	66
Total files in the Project	66	66

To try to get around one of the threats to the validity of that experiment, in which only one developer was used, the second experiment consisted of another case study, this time involving 4 developers, all with one or two years of experience in Android development. Their task was to implement, with and without the framework, a simple project involving three business classes, shown in the diagram of Figure 11.

In this case study, each developer has implemented both projects individually, i.e., there was no collaboration among them in any project. As in the first experiment, each developer first implemented the project without using JustBusiness and, after, using it.

At the end of the second case study, it was observed that, on average, the project developed with JustBusiness decreased approximately 93% of development time, 93% of written lines of code and 96% of files created by the developer, corroborating the results of the previous experiment that indicated gains in productivity when the framework was used. Comparative data for the two developed projects are detailed in Table 4, where the values represent the arithmetic averages of the individual values for each developer.

Table 4: Comparative analysis of JustBusiness and the Traditional Development of the Second Case Study.

Type of Development	JustBusiness	Traditional
Time	50 minutes	743 minutes
Lines written by Developer	256	3935
Files created by Developer	3	79
Total files in the Project	79	79

Finally, we identify as other threats to the validity of the experiments the developers' knowledge level in the Android platform and the difficulty level of the developed projects. Although those factors possibly would imply changes in the values of the items evaluated for each developer, particularly the development time and written lines of code, we believe that the resulting values for the projects that used the framework will be even lower than the ones obtained by the projects without the framework due to the difference in the number of files generated for both cases. We intend to conduct other case studies to prove that hypothesis.

6 CONCLUSION AND FUTURE WORK

This work presented JustBusiness, a framework for developing Android business applications using the Naked Objects architectural pattern. The main benefit of the framework is the automatic generation of user interfaces and CRUD code, thus accelerating the Android application development. Two case studies have been carried out and demonstrated that the use of

the framework promotes a gain in productivity, since it reduces the development time and the number of lines of code and files generated by developers, when compared to solutions that have not used JustBusiness.

Despite its advantages, the framework has some limitations, such as the support to only local data persistence in SQLite database and the lack of customization in the interfaces that were generated automatically.

As future work, we intend to add model-driven developement techniques to the code generation task and provide support for other data types, like images and videos, and other data persistence mechanisms, such as XML and JSON. Additionally, it is intended to introduce validation mechanisms for forms components. Finally, we plan to conduct a study to improve the usability of the generated interfaces.

REFERENCES

(2008). Jmatter. http://jmatter.org/. [Online; accessed 9-July-2015].

(2010a). Apache isis. http://isis.apache.org. [Online; accessed 9-September-2015].

(2010b). Naked object for android. http://sourceforge.net/projects/noforandorid/. [Online; accessed 10-October-2015].

(2010c). Robolectric. http://robolectric.org/. [Online; accessed 13-October-2015].

(2013a). Entities. http://entitiesframework.blogspot.com.br/. [Online; accessed 7-October-2015].

(2013b). Isis android viewer. https://github.com/DImuthuUpe/ISISAndroidViewer/. [Online; accessed 7-August-2015].

(2015a). Naked objects framework. http://www.nakedobjects.org/. [Online; accessed 9-September-2015].

(2015b). Naked objects mvc. http://nakedobjects.net/. [Online; accessed 9-September-2015].

(2015c). Ormlite. http://ormlite.com. [Online; accessed 14-October-2015].

Brandao, M., Cortes, M., and Goncalves, E. (2012). Entities: A framework based on naked objects for development of transient web transientes. In *Informatica (CLEI), 2012 XXXVIII Conferencia Latinoamericana En*, pages 1–10.

Broinizi, M. E. B., Ferreira, J. a. E., and Goldman, A. (2008). Using annotations in the naked objects framework to explore data requirements. In *Proceedings of the 2008 ACM Symposium on Applied Computing*, SAC '08, pages 630–637, New York, NY, USA. ACM.

da Cruz, A. M. R. and Faria, J. P. (2010). A metamodel-based approach for automatic user interface generation. In Petriu, D. C., Rouquette, N., and Haugen, O., editors, *Model Driven Engineering Languages and Systems*, volume 6394 of *Lecture Notes in Computer Science*, pages 256–270. Springer Berlin Heidelberg.

Keranen, H. and Abrahamsson, P. (2005). Naked objects versus traditional mobile platform development: a comparative case study. In *Software Engineering and Advanced Applications, 2005. 31st EUROMICRO Conference on*, pages 274–281.

Milosavljević, B., Vidaković, M., Komazec, S., and Milosavljević, G. (2003). User interface code generation for ejb-based data models using intermediate form representations. In *Proceedings of the 2Nd International Conference on Principles and Practice of Programming in Java*, PPPJ '03, pages 125–132, New York, NY, USA. Computer Science Press, Inc.

Nilsson, E. G. (2009). Design patterns for user interface for mobile applications. *Adv. Eng. Softw.*, 40(12):1318–1328.

Pawson, R. (2004). *Naked Objects*. PhD thesis, University of Dublin, Trinity College.

Pawson, R. and Matthews, R. (2001). Naked objects: A technique for designing more expressive systems. *SIGPLAN Not.*, 36(12):61–67.

Pawson, R. and Matthews, R. (2002). Naked objects. In *Companion of the 17th Annual ACM SIGPLAN Conference on Object-oriented Programming, Systems, Languages, and Applications*, OOPSLA '02, pages 36–37, New York, NY, USA. ACM.

Pawson, R. and Wade, V. (2003). Agile development using naked objects. In *Proceedings of the 4th International Conference on Extreme Programming and Agile Processes in Software Engineering*, XP'03, pages 97–103, Berlin, Heidelberg. Springer-Verlag.

Raja, A. and Lakshmanan, D. (2010). Article: Naked objects framework. *International Journal of Computer Applications*, 1(20):37–41. Published By Foundation of Computer Science.

A Flexible Mechanism for Data Confidentiality in Cloud Database Scenarios

Eliseu C. Branco Jr.[1], Jose Maria Monteiro[2], Roney Reis[2] and Javam C. Machado[2]
[1]*Computer Networks Course, University Center Estacio of Ceara, Fortaleza, Brazil*
[2]*Department of Computer Science, Federal University of Ceara, Fortaleza, Brazil*
eliseu.junior@estacio.br, {monteiro, roneyreis, javam}@lia.ufc.br

Keywords: Data Confidentiality, Cloud Database, Information Decomposition.

Abstract: Cloud computing is a recent trend of technology that aims to provide unlimited, on-demand, elastic computing and data storage resources. In this context, cloud services decrease the need for local data storage and the infrastructure costs. However, hosting confidential data at a cloud storage service requires the transfer of control of the data to a semi-trusted external provider. Therefore, data confidentiality is the top concern from the cloud issues list. Recently, three main approaches have been introduced to ensure data confidentiality in cloud services: data encryption; combination of encryption and fragmentation; and fragmentation. In this paper, we present i-OBJECT, a new approach to preserve data confidentiality in cloud services. The proposed mechanism uses information decomposition to split data into unrecognizable parts and store them in different cloud service providers. Besides, i-OBJECT is a flexible mechanism since it can be used alone or together with other previously approaches in order to increase the data confidentiality level. Thus, a user may trade performance or data utility for a potential increase in the degree of data confidentiality. Experimental results show the potential efficiency of the proposed approach.

1 INTRODUCTION

Cloud Computing moves the application software and databases to large data centers, where data management may not be sufficiently trustworthy. Cloud storage is an increasingly popular class of services for archiving, backup and sharing data. There is an important cost-benefit relation for individuals and small organizations in storing their data using cloud storage services and delegating to them the responsibility of data storage and management (Ciriani et al., 2009). Despite the big business and technical advantages of the cloud storage services, the data confidentiality concern has been one of the major hurdles preventing its widespread adoption.

The concept of privacy varies widely among countries, cultures and jurisdictions. So, a concise definition is elusive if not impossible (Clarke, 1999). For the purposes of this discussion, privacy is "the claim of individuals, groups or institutions to determine for themselves when, how and to what extent the information about them is communicated to others" (Camenisch et al., 2011). Privacy protects access to the person, whereas confidentiality protects access to the data. So, confidentiality is the assurance that certain information that may include a subject's identity, health, lifestyle information or a sponsor's proprietary information would not be disclosed without permission from the subject (or sponsor). When dealing with cloud environments, confidentiality implies that a customer's data and computation tasks are to be kept confidential from both the cloud provider and other customers (Zhifeng and Yang, 2013).

Recently, three main approaches have been introduced to ensure the data confidentiality in cloud environments: a) data encryption, b) combination of encryption and fragmentation (Ciriani et al., 2010), and c) fragmentation (Ciriani et al., 2009). However, in this context, it is in fact crucial to guarantee a proper balance between data confidentiality, on one hand, and other properties, such as, data utility, query execution overhead, and performance on the other hand (Samarati and di Vimercati, 2010; Joseph et al., 2013).

The first approach, denoted by data encryption, consists in encrypting all the data collections. This technique is adopted in the database outsourcing scenario (Ciriani et al., 2010). Actually, encryption algorithms presents increasingly lower costs. Cryptography becomes an inexpensive tool that supports the protection of confidentiality when storing or communicating data (Ciriani et al., 2010). However, deal-

Jr., E., Monteiro, J., Reis, R. and Machado, J.
A Flexible Mechanism for Data Confidentiality in Cloud Database Scenarios.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 359-368
ISBN: 978-989-758-187-8

ing with encrypted data may make query processing more expensive (Ciriani et al., 2009; Ciriani et al., 2010). Some techniques have been proposed to enabling the execution of queries directly on encrypted data (remember that confidentiality demands that data decryption must be possible only at the client side) (Samarati and di Vimercati, 2010). These techniques associate with encrypted data indexing information on which queries can be executed. The mainly challenger for indexing methods is the trade off between precision and privacy: more precise indexes provide more efficient query execution but a greater exposure to possible privacy violations (Ceselli et al., 2005; Samarati and di Vimercati, 2010). Besides, the solutions based on an extensive use of encryption suffer from significant consequences due to loss of keys. In the real scenarios, key management, particularly the operations at the human side, is a hard and delicate process (Samarati and di Vimercati, 2010).

The second approach, called combination of encryption and fragmentation, uses encryption together with data fragmentation. It applies encryption only on the sensitive attributes and splits the attributes with sensitive association into several fragments, which are stored by different cloud storage services (Ciriani et al., 2010). In other words, sensitive association constraints are solved via fragmentation, and encryption is limited to those attributes that are sensitive by themselves. Thus, a single cloud service provider cannot join these fragments for responding queries. Therefore, these techniques must also be accompanied by proper query transformation techniques defining how queries on the original data are translated into queries on the fragmented data. Besides, splitting the attributes with sensitive association into some fragment is a NP-hard problem (Samarati and di Vimercati, 2010; Joseph et al., 2013).

The third approach, denoted by fragmentation, does not use cryptography. In this approach, the sensitive attributes remains under the client's custody while the attributes with sensitive association are split into several fragments, which are stored by different cloud storage services (Ciriani et al., 2009). It is important to note that this approach has the same drawbacks discussed previously (for the combination of encryption and fragmentation approach) regarding to query execution (Samarati and di Vimercati, 2010; Joseph et al., 2013).

In this paper, we present i-OBJECT, a new approach to preserve data confidentiality in cloud storage services. The science behind i-OBJECT uses concepts of the Hegel's Doctrine of Being. The proposed approach is based on the information decomposition to split data into unrecognizable parts and store them in different cloud service providers. Besides, i-OBJECT is a flexible mechanism since it can be used alone or together with other previously approaches in order to increase the data confidentiality level. Thus, a user may trade performance or data utility for a potential increase in the degree of data confidentiality. Experimental results show the potential efficiency of the i-OBJECT.

The remain of this paper is organized as follows. Section 2 presents the proposed approach, called i-OBJECT. Experimental results are presented in Section 3. Next, Section 4 addresses related works. Finally, Section 5 concludes this paper and outlines future works.

2 A DECOMPOSITION-BASED APPROACH FOR DATA CONFIDENTIALITY

The proposed approach for ensuring data confidentiality in cloud environments, denoted i-OBJECT, was designed for transactional data. In this environments, reads are much more frequent than write operations. Thus, i-OBJECT needs to be fast to decompose a file and much faster to recompose a file stored in the cloud.

The i-OBJECT approach was inspired by the German philosopher Hegel's work, according to which an object has three fundamental characteristics (Hegel, 1991): quality, quantity and measure. From this idea, we developed the concept of information object (see Definition 1). From this concept, we have developed the processes to: i) fragment a file in a sequence of information objects and ii) decompose each information object in its properties (quality, quantity and measure).

The i-OBJECT approach has three phases: data fragmentation, decomposition and dispersion, which will be discussed later. Figure 1 shows an overview of the the i-OBJECT approach.

2.1 The Fragmentation Phase

In the fragmentation phase, the basic idea consists in split an input file F in a sequence of n information objects (see Definition 1). Then, we can represent a file F as an ordered set $\{iObj_1, iObj_2, \cdots, iObj_n\}$ of i-OBJECTs.

Definition 1 (i-OBJECT)**.** *An information object, i-OBJECT for short, is a piece of 256 sequential bytes from a file.* ◇

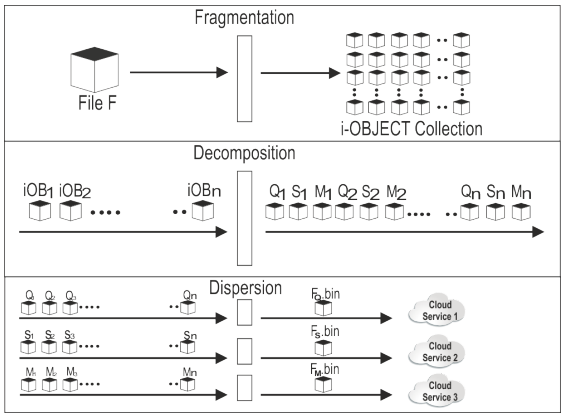


Figure 1: i-OBJECT Approach Overview.

2.2 The Decomposition Phase

The decomposition phase receives as input a file F, represented as an ordered set $\{iObj_1, iObj_2, \cdots, iObj_n\}$ of i-OBJECTs, and using the Hegel's theory (Hegel, 1991), according to which an object has three fundamental characteristics (quality, quantity and measure), decomposes F in three files: Fq.bin, Fs.bin and Fm.bin, which represent, respectively, F's quality, quantity and measure.

In order to understand the decomposition phase, it is necessary to formally define the quality, quantity and measure properties. These definitions are presented next.

Definition 2 (Quality)**.** *Quality is the set of diverse bytes that composes a particular i-OBJECT. Let $iObj_k$ be an i-OBJECT, $Q(iObj_k)$ denotes the quality property of the $iObj_k$. $Q(iObj_k)$ is a ordered vector containing the* m *diverse bytes present in $iObj_k$. More formally, $Q(iObj_k) = \{b_1, b_2, b_3, \cdots, b_m\}$ such that $1 \leqslant b_i \leqslant 256$ and $i \neq j \rightarrow b_i \neq b_j$, where b_i is a byte present in $iObj_k$.* ◇

Definition 3 (Quantity)**.** *Quantity is an array containing the number of times that each distinct byte appears in a specific i-Object. Let $iObj_k$ be an i-OBJECT, $S(iObj_k)$ denotes the quantity property of the $iObj_k$. $S(iObj_k)$ is a vector containing, for each different byte b_j (representing a ASCII Symbol) present in $Q(iObj_k)$ the number of times that b_i appears in $iObj_k$. More formally, $S(iObj_k) = \{s_1, s_2, s_3, \cdots, s_m\}$ such that $1 \leqslant s_i \leqslant 256$, where s_i represents the number of times that b_i appears in $iObj_k$.* ◇

Definition 4 (Measure)**.** *Measure is a two-dimensional array containing, for each diverse byte that composes a particular i-OBJECT, a vector with the positions where this byte occurs in the i-OBJECT. Let $iObj_k$ be an i-OBJECT, $M(iObj_k)$ denotes the measure property of the $iObj_k$. $M(iObj_k)$ is a two-dimensional array containing, for each different byte b_j present in $Q(iObj_k)$, an array m_{b_j} storing the positions in which the byte b_j appears in $iObj_k$. More formally, $M(iObj_k) = \{m_{b_1}, m_{b_2}, \cdots, m_{b_m}\}$, such that, $m = 256$ and $1 \leqslant size(m_{b_i}) \leqslant 256$.* ◇

Given a file F, where $F = \{iObj_1, iObj_2, \cdots, iObj_n\}$. Initially, the decomposition phase consists in extracting, for each i-OBJECT $iObj_k$, where $1 \leqslant k \leqslant n$ and $iObj_k \in F$, its properties: quality ($Q(iObj_k)$), quantity ($S(iObj_k)$) and measure ($M(iObj_k)$).

Next, the proposed approach combines the quality values for all i-OBJECTs in the set $\{iObj_1, iObj_2, \cdots, iObj_n\}$ and creates a file called Fq.bin. After this, the i-OBJECT approach combines the quantity values for all i-OBJECTs in the set $\{iObj_1, iObj_2, \cdots, iObj_n\}$ and creates a file denoted by Fs.bin. Finally, the proposed approach combines the measure values for all i-OBJECTs in the set $\{iObj_1, iObj_2, \cdots, iObj_n\}$ and creates a file called Fm.bin (see Figure 1).

In order to illustrated how an i-OBJECT $iObj_k$ is decomposed into its three basic properties (Q, S and M) consider the following example, denoted Example 1. **Example 1:** Consider an i-OBJECT $iObj_k$ containing the following text:

```
"Google dropped its cloud computing prices
and other vendors are expected to follow
suit, but the lower pricing may not be
the key for attracting enterprises to the
cloud. When Enterprises comes to cloud,
they're more concerned about privacy
and security"
```

Note that $iObj_k$ contains 31 diverse bytes, where each byte represents a ASCII symbol. So, the quality property of the $iObj_k$ is a vector with 31 elements, as follows: $Q(iObj_k)$ = { 32(space), 34("), 39('), 44(,) 46(.), 69(E), 71(G), 87(W), 97(a), 98(b), 99(c), 100(d), 101(e), 102(f), 103(g), 104(h), 105(i), 107(k), 108(l), 109(m), 110(n), 111(o), 112(p), 114(r), 115(s), 116(t), 117(u), 118(v), 119(w), 120(x), 121(y) } Thereby, the quantity property of the $iObj_k$ is also a vector with 31 elements: $S(iObj_k)$ = { 40, 2, 1, 2, 1, 1, 1, 1, 8, 3, 13, 10, 28, 2, 4, 6, 11, 1, 7, 4, 12, 21, 9, 18, 10, 21, 8, 2, 2, 1, 5 } It's important to note the relationship between quality and quantity properties. Note that, for example, the character "space" (ASCII 32), the first element in $Q(iObj_k)$, denoted by $Q(iObj_k)_1$, appears 40 times in the $iObj_k$, and the character "y" (ASCII 121), the last element in $Q(iObj_k)$, denoted by $Q(iObj_k)_{31}$, occurs 5 times in the $iObj_k$. In this

scenario, the measure property of the $iObj_k$ is a two-dimensional array containing 31 arrays, as follows: $M(iObj_k)$ = {{ 7, 15, 19, 25, 35, 42, 46, 52, 60, 64, 73, 76, 83, 89, 93, 97, 103, 111, 115, 119, 122, 126, 130, 134, 145, 157, 160, 164, 171, 176, 188, 194, 197, 204, 212, 217, 227, 233, 245, 241 }, { 0, 255}, { 209 }, ⋯, { 114, 129, 208, 253, 240 }} Observe that, for example, the character "y" (ASCII 121), the 31st element in $Q(iObj_k)$, occurs 5 times ($Q(iObj_k)_{31} = 5$) in the $iObj_k$, in the positions 114, 129, 208, 253 and 240, which are represented by the last array in $M(iObj_k)$.

The Algorithm 1 illustrates how a file F is decomposed in the files $F_q.bin$, $F_s.bin$ and $F_m.bin$. The Algorithm 2 shows how the files $F_q.bin$, $F_s.bin$ and $F_m.bin$ are used to recompose the file F.

The decomposition algorithm (Algorithm 1) is performed in two stages. The first step (lines 1 to 17) obtains the information of the bytes ordinal positions (M) of $iObj_k$ (Q) and stores it in a two-dimensional vector temp [256] [256] (line 12), where the first dimension of the vector represents the decimal value of the byte and the second dimension is a list of the positions occupied by the byte's occurrence in $iObj_k$. The second step of the process (lines 18 to 41) generates three vectors containing the Q and S information, where each element of these sets is represented by one bit, and a byte vector M, which contains the positions grouped in ascending order of Q elements. In groups with more than one element, the two last elements are made to reverse its positions to indicate the end of the group.

The recomposition algorithm (Algorithm 2) receives as input the files $F_q.bin$, $F_s.bin$ and $F_m.bin$ and restores the original file F. The files are read sequentially in blocks of 256 bits (Q and S) and 256 bytes (M) (lines 4 to 6) so that the rebuilding of elements Q, S and M starts. The algorithm goes through the sample space of Q elements (0 to 255), identifying bytes exist in $iObj_k$ (line 11) and if they occur one time or more than once (S) (line 15) to then retrieve positions occupied by these bytes and insert them in vector $iObj_k$ (rows 18 and 22).

Algorithm 1: Decomposition function.

```
   input  : File F
   output : File Fq.bin, Fs.bin, Fm.bin
 1 Function − Decomposition(F);
 2 begin
 3     IObj = Read(F,256);
 4     CreateFile(Fq.bin,Fs.bin,Fm.bin);
 5     Temp[256][256];
 6     while IObj not null do
 7         byte = 0; cont = 0; freq[] = 0;
 8         for pos = 0 to 255 do
 9             byte = IObj[pos];
10             freq[byte] ++;
11             cont = freq[byte];
12             Temp[byte][cont] = pos;
13         end for
14         CreateQSM(Temp,freq);
15         IObj = Read(F,256);
16     end while
17 end
18 Function − CreateQSM(Temp[256][256],freq[256]);
19 begin
20     Q[256] = 0;S[256] = 0;M[256] = 0;
21     pos = 0; postemp = 0; cont = 0; cont2 = 0;
22     for byte = 0 to 255 do
23         if freq[byte] ≥ 1 then
24             Q[byte] = 1;
25             if freq[byte] ≥ 2 then
26                 S[cont2] = 1;
27                 postemp = Temp[byte][freq[byte]];
28                 Temp[byte][freq[byte]] =
                     Temp[byte][freq[byte] − 1];
29                 Temp[byte][freq[byte] − 1] = postemp;
30             end if
31             for cont = 1 to freq[byte] do
32                 M[pos] = Temp[byte][cont];
33                 pos ++;
34             end for
35             cont2 ++;
36         end if
37     end for
38     Append(Fq.bin,Q);
39     Append(Fs.bin,S);
40     Append(Fm.bin,M);
41 end
```

2.3 The Dispersion Phase

In the dispersion step, the files $F_q.bin$, $F_s.bin$ and $F_m.bin$ are spread across different cloud storage service providers. So, i-OBJECT requires that these three files are isolated between themselves.

3 DATA CONFIDENTIALITY CONSIDERATIONS

The data confidentiality in the proposed approach stems from the fact that the files Fq.bin, Fs.bin and Fm.bin are stored in different cloud providers, which are physically and administratively independent. According to (Resch and Plank, 2011), physical data dispersion in different storage servers, along with the careful choice of the number of servers and the amount of fragments needed for restore the original

Algorithm 2: Recomposition function.

```
   input  : File Fq.bin, Fs.bin, Fm.bin
   output : File F
 1 Function − Recomposition(F);
 2 begin
 3     CreateFile(F);
 4     Q[256] = Read(Fq.bin,256);
 5     S[256] = Read(Fs.bin,256);
 6     M[256] = Read(Fm.bin,256);
 7     while Q not null do
 8         ordem = 0; cont = 0; pos = 0;
 9         for bit = 0 to 255 do
10             if Q[bit] ≠ 0 then
11                 pos ← M[ordem];
12                 IObj[pos] ← CodASCII(Q[bit]);
13                 ordem ++;
14                 if S[cont] = 1 then
15                     while M[ordem] ⩾ pos do
16                         pos ← M[ordem];
17                         IObj[pos] ← CodASCII(Q[bit]);
18                         ordem ++;
19                     end while
20                     pos ← M[ordem];
21                     IObj[pos] ← CodASCII(Q[bit]);
22                     ordem ++;
23                 end if
24                 cont ++;
25             end if
26         end for
27         Append(F,IObj);
28         Q[256] = Read(Fq.bin,256);
29         S[256] = Read(Fs.bin,256);
30         M[256] = Read(Fm.bin,256);
31     end while
32 end
```

files, reduces the chances of an attacker and is enough to make a system safe. So, in i-OBJECT, the degree of data confidentiality is based on the difficulty of the attackers to reconstruct the i-OBJECTS from one of these three files, Fq.bin, Fs.bin or Fm.bin. Besides, i-OBJECT is a flexible mechanism since it can be used alone or together with other previously approaches (such as encryption algorithms like AES, DES or 3-DES) in order to increase the data confidentiality level.

4 EXPERIMENTAL EVALUATION

In order to show the potentials of i-OBJECT, several experiments have been conducted. The main results achieved so far are presented and discussed in this section. Thus, we first provide information on how the experimentation environment was set up. Thereafter, empirical results are quantitatively presented and qualitatively discussed.

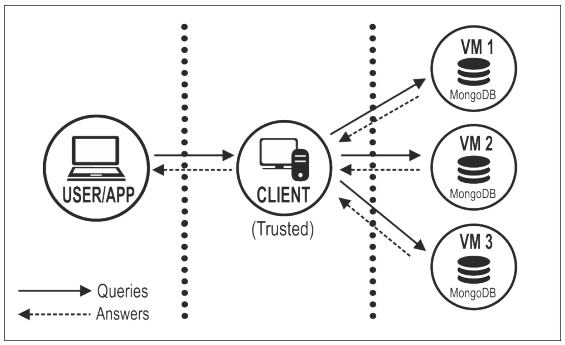


Figure 2: Experimental Architecture.

4.1 Experimental Setup

We implemented i-OBJECT and the other data confidentiality approaches using C and Java. In order to run these approaches we have used a private cloud computing infrastructure based on OpenStack. Figure 2 shows the architecture used in the experiments, which contains two kinds of virtual machines: the client node and the data storage nodes. The client, a Trusted Third Party (TTP), runs the i-OBJECT algorithms: decomposition and recomposition. The data storage nodes (called VM1, VM2 and VM3) emulate three different cloud storage service providers. We assume that the data nodes provide reliable content storage and data management but are not trusted by the client to maintain data privacy.

Each data storage node has the following configuration: Ubuntu 14.04 operating system, Intel Xeon 2.20 GHz processor, 4 GB memory and 50 GB disk. The client is a Intel Xeon 2.20 GHz processor, 4 GB memory and 40 GB disk capacity, running a Windows Server 2008.

Besides this, each data storage node has a MongoDB instance with default settings. MongoDB is an open source, scalable, high performance, schema-free, document oriented database. We opted to use the MongoDB because it is one of the most used database in cloud computing environments and exists opportunities for improvement its security and privacy. MongoDB supports a binary wire-level protocol but doesn't provide a method to automatically encrypt data. This means that any attacker with access to the file system can directly extract the information from the files (Okman et al., 2011).

In order to evaluate the i-OBJECT efficiency, we have used a document collection, synthetically created, which contains files (documents) with different sizes. Each file has four parts (or attributes), which have the same size. These attributes are: curriculum vitae (A1), paper text (A2), author photo (A3) and paper evaluation (A4).

4.2 Test Results

In this section, we present the results of the experiments we carried out. For evaluate the i-OBJECT efficiency, we have used two metrics: the input and output times. The input time is defined as the total amount of time spent to process a file F and generates the data that will be send to the cloud storage service providers. The output time is defined as the total amount of time spent to process the data received from the cloud storage service providers in order to remount the original file F. It is important to highlight that the time spend in the communication process, to send and receive data to the cloud providers do not composes neither the input nor the output time. We have evaluated different file sizes (2^{18}, 2^{20} and 2^{22} bytes). For each distinct file size, we have used 10 files and computes the average time for decomposes and recomposes these files. To validate the i-OBJECT approach, we have evaluated four different scenarios, which will be discussed next.

4.2.1 Scenario 1: Encryption Algorithms

The first scenario was running with the aim of compare the most popular symmetric cryptographic algorithms: AES, DES and 3-DES. It is important to note that, in this experiment, the Input Time matches the Encryption Time (the spent time to encrypt a file F) and Output Time matches the Decryption Time (time necessary to decrypt a file F).

So, in this scenario, the client receives a file F from the user, encrypt it, generating a new file F_e, and sends F_e to VM1. Figure 3 shows the encryption time for the algorithms AES, DES and 3-DES. Next, the client receives the encrypted file F_e from VM1, decrypt it, generating the original file F and sends F to the user. Figure 4 shows the decryption time for the algorithms AES, DES and 3-DES. Note that, for files with size of 2^{16} bytes these three algorithms presented the same encryption time, while AES and DES presented the same decryption time. However, for files with sizes of 2^{18}, 2^{20}, and 2^{30} bytes AES outperforms DES and 3-DES, for both encryption and decryption.

4.2.2 Scenario 2: Data Confidentiality Approaches

In the second scenario, we compared the i-OBJECT approach with the three main approaches to ensure the data confidentiality in cloud environments: a) data encryption, b) combination of encryption and fragmentation, and c) fragmentation (see Section 1) (Samarati and di Vimercati, 2010; Joseph et al., 2013).

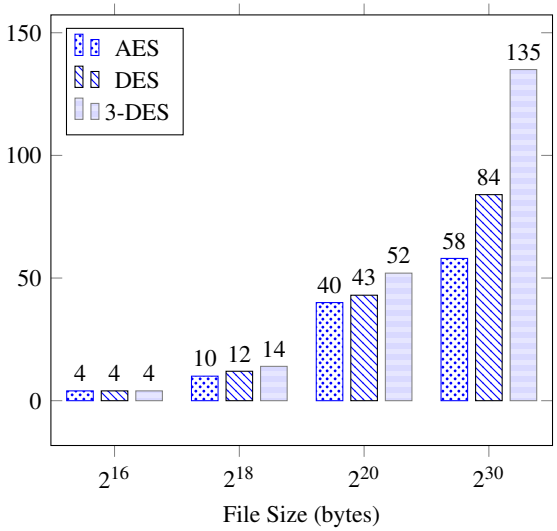


Figure 3: Scenario 1: Encryption Time.

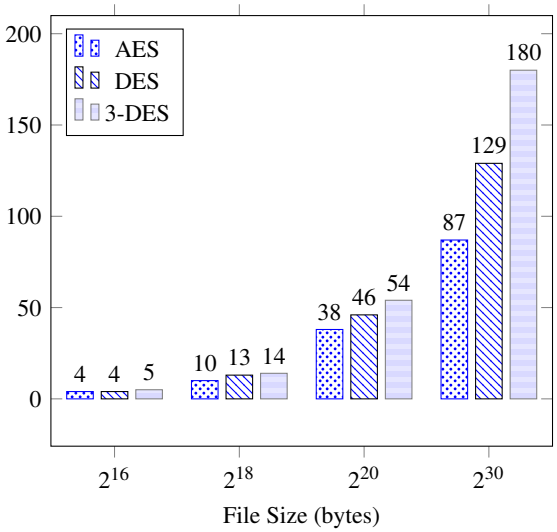


Figure 4: Scenario 1: Decryption Time.

In order to run the approaches (b), combination of encryption and fragmentation, and (c), fragmentation, it is necessary to define which attributes are sensitive, besides to identify the sensitive association between attributes. Moreover, splitting the attributes with sensitive association into some fragment is a NP-hard problem (Samarati and di Vimercati, 2010; Joseph et al., 2013).

Thus, we have assumed that each document has four attributes: curriculum vitae (A1), paper text (A2), author photo (A3) and paper evaluation (A4). Besides, we supposed that there is a set C of sensitive association constraints, with the following constraints: C_1={A1}, C_2 = {A2}, C_3 = {A2, A4}, C_4 = {A1, A3}. So, the attributes A_1 and A_3 are considered sensitive and must be encrypted in approaches

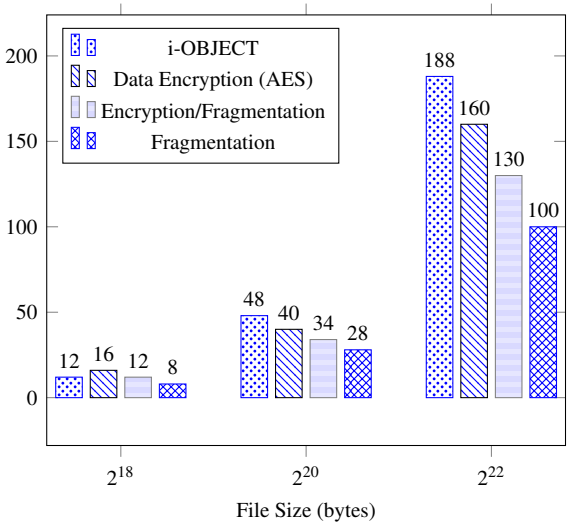


Figure 5: Scenario 2: Input Time.

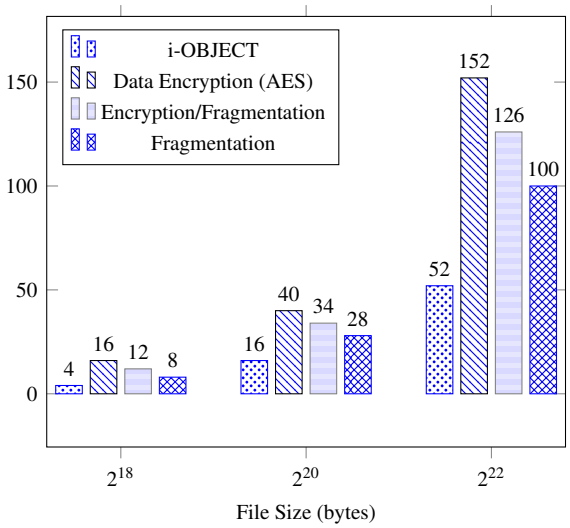


Figure 6: Scenario 2: Output Time.

(b) and (c). The constraint $C_3 = \{A2, A4\}$ indicates that there is a sensitive association between A2 and A4. The constraint $C_4 = \{A1, A3\}$ indicates that there is a sensitive association between A1 and A3, and these attributes should be stored in different servers in the cloud. Based on the set C of confidentiality constraints, a set P of data fragments was generated, as following: i) the approach (b), combination of encryption and fragmentation, produced the fragments P_1={A1,A4} and P_2 = {A2,A3}; and ii) the approach (c), fragmentation, formed the fragments P_3 = {A1, A3}, P_4 = {A2} and P_5 = {A4}.

It is important to emphasize that the time necessary to define the set of fragments (fragments schema) for splitting the attributes with sensitive association, that is a NP-hard problem, was not considered in this experiment. Furthermore, for the approach (a), data encryption, we have used the AES algorithm, since it presented best results in the first scenario.

In this experiment, we considered performance with respect to the following metrics: (i) Input Time and (ii) Output Time. These metrics change a little according to the used data confidentiality approach.

Input Time is computed as following:

- Approach (a), data encryption: time to encrypt a file F using AES, generating a file F_e. The file F_e will be send to VM1;
- Approach (b), combination of encryption and fragmentation: time to encrypt A_1 and A_3, plus the time to generates P_1 and P_2. Where, A_1 and P_1 will be send to VM1, while A_3 and P_2 will be send to VM2.
- Approach (c), fragmentation: the time to generates P_3, P_4 and P_5. Where, P_3 will be send to VM1, P_4 to VM2 and P_5 to VM3.
- i-OBJECT Approach: time to decompose a file F into $F_q.bin$, $F_s.bin$ and $F_m.bin$. Where, $F_q.bin$ will be send to VM1, $F_s.bin$ to VM2 and $F_m.bin$ to VM3;

Output Time is computed as following:

- Approach (a), data encryption: time to decrypt a file F_e using AES, generating the original file F;
- Approach (b), combination of encryption and fragmentation: time to decrypt A_1 and A_3, plus the time to join A_1, A_3, P_1 and P_2 in order to remount the file F;
- Approach (c), fragmentation: the time to join P_3, P_4, P_5 and the sensitive attributes stored in the client;
- i-OBJECT Approach: time to recompose a file F from $F_q.bin$, $F_s.bin$ and $F_m.bin$.

Figure 5 shows the input time for the evaluated approaches. Note that i-OBJECT approach has a performance slightly worse than Data Encryption (AES). Fragmentation approach outperforms the other approaches, for all file sizes. On the other hand, the last two approaches, Encryption/Fragmentation and Fragmentation, need to define the set of fragments (fragmentation schema) for splitting the attributes with sensitive association. However, how we have used a fixed example, the time necessary to define the fragmentation schema was not computed. In part, this explains the better results obtained by these two approaches.

Figure 6 shows the output time for the evaluated approaches. Note that i-OBJECT outperforms all the other approaches, for all file sizes. It is important to

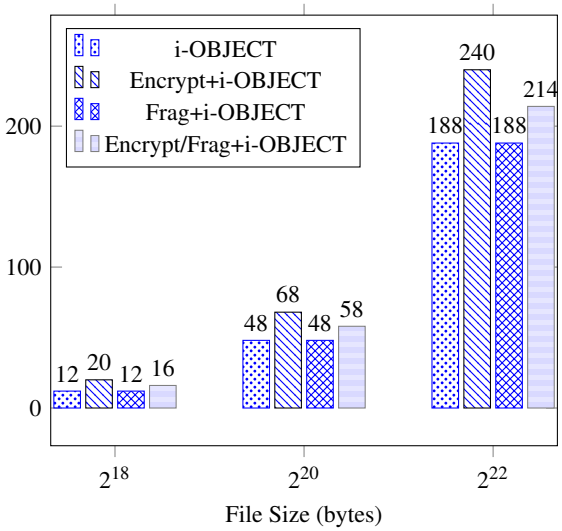


Figure 7: Scenario 3: Input Time.

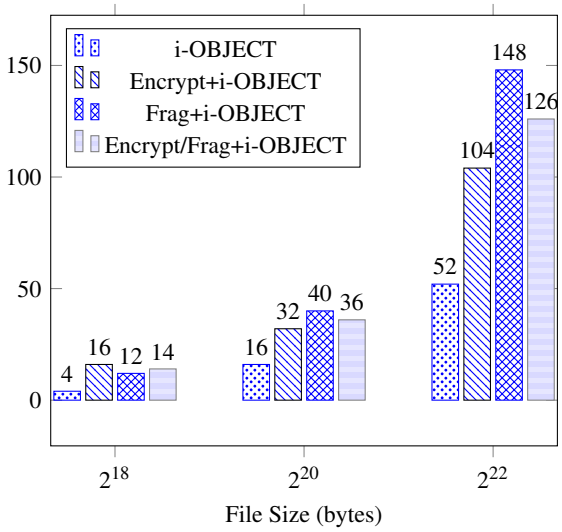


Figure 8: Scenario 3: Output Time.

highlight, for the file size of 2^{22} bytes, i-OBJECT is 88s slower than the Fragmentation approach in the input phase, that is, to process and a file F before sending it to the cloud storage service provider. However, for the same file size, i-OBJECT is 48s faster than the Fragmentation approach. So, for a complete cycle of file write and read, i-OBJECT is just 40s slower than the Fragmentation approach. Then, if the user writes F one time and reads F two times, i-OBJECT is 8s faster than Fragmentation approach. Thus, i-OBJECT outperforms all the other previous approach in scenarios where the number of reads is at least twice larger than the number of writes, which is expected real databases and cloud storage environments.

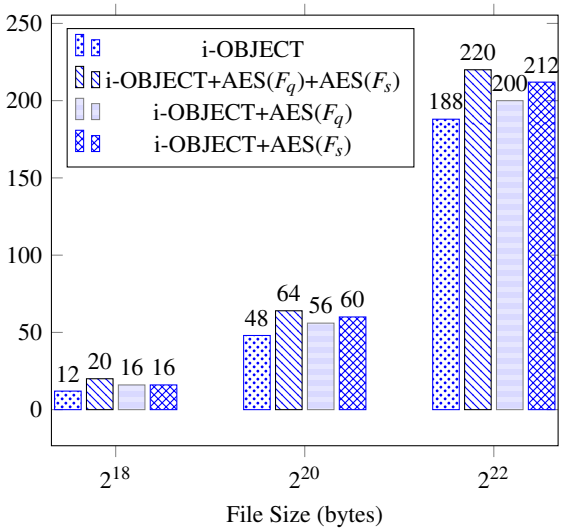


Figure 9: Scenario 4: Input Time.

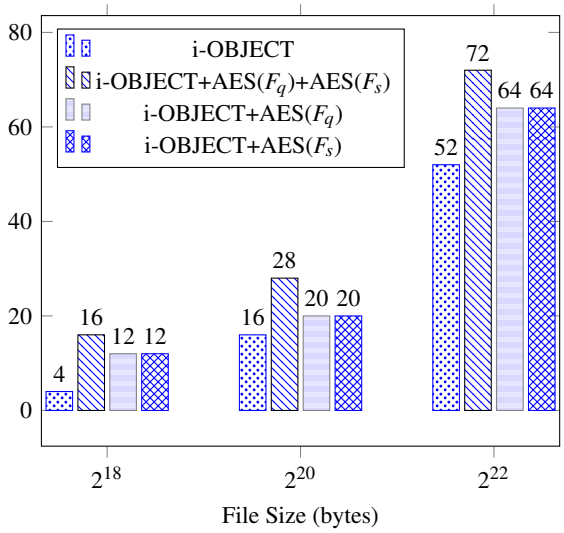


Figure 10: Scenario 4: Output Time.

4.2.3 Scenario 3: Using i-OBJECT Together with Previous Approaches

In the third scenario, we evaluated the use of i-OBJECT together with previous approaches. We believe that i-OBJECT can be used together with other data confidentiality approaches in order to improve their data confidentiality levels.

Figure 7 shows the input time for the evaluated approaches. In this chart, the first bar shows the input time to i-OBJECT (that is, the time to decompose a file F); the second bar represents the input time to apply the Data Encryption approach and, after that, the i-OBJECT (that is, the time to encrypt a file F, producing a new file F_e, plus the time to decompose

F_e); the third bar indicates the input time to apply the Encryption/Fragmentation approach and, next, the i-OBJECT; finally, the fourth bar denotes the input time to apply the Fragmentation approach and then the i-OBJECT. Then, we can argue that i-OBJECT is a flexible approach, in the sense that it can be used together with previous approaches, in order to improve their data confidentiality level. The results presented in Figure 7 show that this strategy provides a small increase in the input time, while providing a high gain in the data confidentiality. Figure 8 shows that the output time overhead has a similar behavior that input time.

4.2.4 Scenario 4: Improving Data Confidentiality in i-OBJECT

In the fourth scenario, we evaluated some strategies to improve the data confidentiality in the i-OBJECT approach. The first strategy consists in encrypt the files $F_q.bin$ and $F_s.bin$, the second strategy consists in encrypt only the file $F_q.bin$ and the third strategy consists in encrypt just the file $F_s.bin$.

Figure 9 and Figure 10 show, respectively. the input and output time with and without the use of these strategies. Note that the strategy of encrypting just the file $F_q.bin$ provides a low overhead, while greatly increases the data confidentiality level of the i-OBJECT approach.

4.2.5 Storage Space Considerations

In the i-OBJECT approach, a file F is decomposed into three files (Fq.bin, Fs.bin and Fm.bin), which are dispersed (sent) to three different cloud providers. The experimental results showed that the size of these files represent, respectively, 12.5%, 12.5% and 90% of the original file size. So, adding these values, the propposed approach provides an overhead of 15% in the disk space utilization. This drawback is minimized since the cloud storage services are designed to store a large quantity of data.

5 RELATED WORK

A significant amount of research has recently been dedicated to the investigation of data confidentiality in cloud computing environment. Most of this work has assumed the data to be entirely encrypted, focusing on the design of queries execution techniques (Ciriani et al., 2010). In (Ceselli et al., 2005) the authors discuss different strategies for evaluating the inference exposure for encrypted data enriched with indexing information, showing that even a limited number of indexes can greatly favor the task for an attacker wishing to violate the data confidentiality provided by encryption.

The first proposal proposing the storage of plaintext data, while ensuring a series of privacy constraints was presented in (Aggarwal, 2005). In this work, the authors suppose data to be split into two fragments, stored on two honest-but-curious service providers, which never exchange information, and resorts to encryption any time these two fragments are not sufficient for enforcing confidentiality constraints. In (Ciriani et al., 2009; Ciriani et al., 2010), the authors address these issues by proposing a solution that first split the data to be protected into several (possibly more than two) different fragments in such a way to break the sensitive associations among attributes and to minimize the amount of attributes represented only in encrypted format. The resulting fragments may be stored at different servers. The proposed heuristic to design these fragments present a polynomial-time computation cost while is able to retrieve solutions close to optimum. In (Xu et al., 2015), the authors propose an efficient graph search based method for the fragmentation problem with confidentiality constraints, which obtains near optimal designs.

The work presented in (Ciriani et al., 2009) proposes a novel paradigm for preserving data confidentiality in data outsourcing which departs from encryption, thus freeing the owner from the burden of its management. The basic idea behind this mechanism is to involve the owner in storing the sensitive attributes. Besides, for each sensitive association, the owner should locally store at least an attribute. The remainder attributes are stored, in the clear, at the server side. With this fragmentation process, an original relation R is then split into two fragments, called F_o and F_s, stored at the data owner and at the server side, respectively. (Wiese, 2010) extends the "vertical fragmentation only" approach and proposes use horizontal fragmentation to filter out confidential rows to be securely stored at the owner site. (Krishna et al., 2012) proposes an approach based on data fragmentation using graph-coloring technique wherein a minimum amount of data is stored at the owner. In (Rekatsinas et al., 2013) the authors present SPARSI, a theoretical framework for partitioning sensitive data across multiple non-colluding adversaries. They introduce the problem of privacy-aware data partitioning, where a sensitive dataset must be partitioned among k untrusted parties (adversaries). The goal is to maximize the utility derived by partitioning and distributing the dataset, while minimizing the total amount of sensitive information disclosed. Solving

privacy-aware partitioning is, in general, NP-hard, but for specific information disclosure functions, good approximate solutions can be found using relaxation techniques.

In (Samarati and di Vimercati, 2010) the authors discuss the main issues to be addressed in cloud storage services, ranging from data confidentiality to data utility. They show the main research directions being investigated for providing effective data confidentiality and for enabling their querying. The survey presented in (Joseph et al., 2013) addressed some approaches for ensuring data confidentiality in untrusted cloud storage services. In (Samarati, 2014), the authors discuss the problems of guaranteeing proper data security and privacy in the cloud, and illustrate possible solutions for them.

6 CONCLUSION AND FUTURE WORK

Experimental results showed the efficiency of i-OBJECT, which can be used with any kind of file and is more suitable for files larger than 256 bytes, files with high entropy and environments where the number of read operations exceeds the number of writes. As a future work, we intend realize a detailed analysis of the i-OBJECT security and evaluate i-OBJECT performance with other data types and using different cloud configurations, including public and mixed clouds.

REFERENCES

Aggarwal, C. C. (2005). On k-anonymity and the curse of dimensionality. In *Proceedings of the 31st international conference on Very large data bases*, pages 901–909. VLDB Endowment.

Camenisch, J., Fischer-Hbner, S., and Rannenberg, K. (2011). *Privacy and identity management for life*. Springer.

Ceselli, A., Damiani, E., De Capitani di Vimercati, S., Jajodia, S., Paraboschi, S., and Samarati, P. (2005). Modeling and assessing inference exposure in encrypted databases. *ACM Transactions on Information and System Security (TISSEC)*, 8(1).

Ciriani, V., De Capitani Di Vimercati, S., Foresti, S., Jajodia, S., Paraboschi, S., and Samarati, P. (2009). Keep a few: Outsourcing data while maintaining confidentiality. In *Proceedings of the 14th European Conference on Research in Computer Security*, ESORICS'09, pages 440–455, Berlin, Heidelberg. Springer-Verlag.

Ciriani, V., Vimercati, S. D. C. D., Foresti, S., Jajodia, S., Paraboschi, S., and Samarati, P. (2010). Combining fragmentation and encryption to protect privacy in data storage. *ACM Trans. Inf. Syst. Secur.*, 13(3):22:1–22:33.

Clarke, R. (1999). Introduction to dataveillance and information privacy, and definition of terms.

Hegel, G. (1991). The encyclopedia logic (tf geraets, wa suchting, hs harris, trans.). *Indianapolis: Hackett*, 1.

Joseph, N. M., Daniel, E., and Vasanthi, N. A. (2013). Article: Survey on privacy-preserving methods for storage in cloud computing. *IJCA Proceedings on Amrita International Conference of Women in Computing - 2013*, AICWIC(4):1–4. Full text available.

Krishna, R. K. N. S., Sayi, T. J. V. R. K. M. K., Mukkamala, R., and Baruah, P. K. (2012). Efficient privacy-preserving data distribution in outsourced environments: A fragmentation-based approach. In *Proceedings of the International Conference on Advances in Computing, Communications and Informatics*, ICACCI '12, pages 589–595, New York, NY, USA. ACM.

Okman, L., Gal-Oz, N., Gonen, Y., Gudes, E., and Abramov, J. (2011). Security issues in nosql databases. In *Trust, Security and Privacy in Computing and Communications (TrustCom), 2011 IEEE 10th International Conference on*, pages 541–547.

Rekatsinas, T., Deshpande, A., and Machanavajjhala, A. (2013). Sparsi: Partitioning sensitive data amongst multiple adversaries. *Proc. VLDB Endow.*, 6(13):1594–1605.

Resch, J. K. and Plank, J. S. (2011). Aont-rs: blending security and performance in dispersed storage systems. In *Proceedings of FAST-2011: 9th Usenix Conference on File and Storage Technologies,February 2011*.

Samarati, P. (2014). Data security and privacy in the cloud. In *Information Security Practice and Experience - 10th International Conference, ISPEC 2014, Fuzhou, China, May 5-8, 2014. Proceedings*, pages 28–41.

Samarati, P. and di Vimercati, S. D. C. (2010). Data protection in outsourcing scenarios: Issues and directions. In *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*, ASIACCS '10, pages 1–14, New York, NY, USA. ACM.

Wiese, L. (2010). *Horizontal fragmentation for data outsourcing with formula-based confidentiality constraints*, pages 101–116. Springer.

Xu, X., Xiong, L., and Liu, J. (2015). Database fragmentation with confidentiality constraints: A graph search approach. In *Proceedings of the 5th ACM Conference on Data and Application Security and Privacy*, CODASPY '15, pages 263–270, New York, NY, USA. ACM.

Zhifeng, X. and Yang, X. (2013). Security and privacy in cloud computing. *Communications Surveys & Tutorials, IEEE*, 15(2):843–859.

Investigating the Use of a Contextualized Vocabulary in the Identification of Technical Debt: A Controlled Experiment

Mário André de Freitas Farias[1,2], José Amancio Santos[3], André Batista da Silva[4], Marcos Kalinowski[5], Manoel Mendonça[2] and Rodrigo Oliveira Spínola[6,7]
[1]*Federal Institute of Sergipe, Lagarto, Sergipe, Brazil*
[2]*Federal University of Bahia, Salvador, Bahia, Brazil*
[3]*State University of Feira de Santana, Feira de Santana, Bahia, Brazil*
[4]*Federal University of Sergipe, Aracaju, Sergipe, Brazil*
[5]*Fluminense Federal University, Rio de Janeiro, Brazil*
[6]*Fraunhofer Proj. Center at UFBA, Salvador, Bahia, Brazil*
[7]*Salvador University, Salvador, Bahia, Brazil*
mario.andre@ifs.edu.br, {zeamancio, andrebtsa}@gmail.com, kalinowski@ic.uff.br, {manoel.g.mendonca, rodrigoospinola}@gmail.com

Keywords: Contextualized Vocabulary, Technical Debt, Code Comment, Controlled Experiment.

Abstract: In order to effectively manage technical debt (TD), a set of indicators has been used by automated approaches to identify TD items. However, some debt may not be directly identified using only metrics collected from the source code. CVM-TD is a model to support the identification of technical debt by considering the developer point of view when identifying TD through code comment analysis. In this paper, we analyze the use of CVM-TD with the purpose of characterizing factors that affect the accuracy of the identification of TD. We performed a controlled experiment investigating the accuracy of CVM-TD and the influence of English skills and developer experience factors. The results indicated that CVM-TD provided promising results considering the accuracy values. English reading skills have an impact on the TD detection process. We could not conclude that the experience level affects this process. Finally, we also observed that many comments suggested by CVM-TD were considered good indicators of TD. The results motivate us continuing to explore code comments in the context of TD identification process in order to improve CVM-TD.

1 INTRODUCTION

The Technical Debt (TD) metaphor reflects the challenging decisions that developers and managers need to take in order to achieve short-term benefits. These decisions may not cause an immediate impact on the software, but may negatively affect the long-term health of a software system or maintenance effort in the future (Izurieta *et al.,* 2012). The metaphor is easy to understand and relevant to both technical and nontechnical practitioners (Alves *et al.,* 2016) (Ernst *et al.,* 2015). In this sense, its acceptance and use have increased in software engineering researches.

In order to effectively manage TD, it is necessary to identify TD items[1] in the project (Guo et *al*., 2014). (Li *et al.,* 2014), in a recent systematic review, reported that code quality analysis techniques have been frequently studied to support the identification of TD. Automatic analysis tools have used software metrics extracted from the source code to identify TD items by comparing values of software metrics to predefined thresholds (Mendes *et al.,* 2015). Although these techniques have shown useful to support the automated identification of some types of debt, they do not cover human factors (e.g., tasks commented as future work) (Zazworka *et al.,* 2013) (Potdar and Shihab, 2014). Thus, large amounts of TD that are undetectable by tools may be left aside. In this sense, pieces of code that need to be refactored to improve the quality of the software may continue unknown. In order to complement the TD identification with more contextual and qualitative data, human factors and the developers' point of view should be considered (Farias *et al.,* 2015).

[1] The term "TD item" represents an instance of Technical Debt.

Farias, M., Santos, J., Silva, A., Kalinowski, M., Mendonça, M. and Spínola, R.
Investigating the Use of a Contextualized Vocabulary in the Identification of Technical Debt: A Controlled Experiment.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 369-378
ISBN: 978-989-758-187-8

In this context, (Potdar and Shihab, 2014) have focused on code comments aiming to identify TD items. Therefore, they manually read more than 101K code comments to detect those that indicate a self-admitted TD. These comments were analyzed in order to identify text patterns that indicate a TD. In the same way, (Maldonado and Shihab, 2015) have read 33K code comments to identify different types of debt using the indicators proposed by (Alves *et al.,* 2014). According to the authors, these patterns can be used to manually identify TD that exists in the project by reading code comments. However, it is hard to perform such a large manual analysis in terms of effort and the process is inherently error prone.

(Farias *et al.,* 2015) presented a Contextualized Vocabulary Model for identifying TD (CVM-TD) based in code comments. CVM-TD uses word classes and code tags to provide a set of TD terms/patterns of comment (a contextualized vocabulary) that may be used to filter comments that need more attention because they may indicate a TD item. CVM-TD was evaluated through an exploratory study on two large open sources projects, jEdit and Lucene, with the goal of characterizing its feasibility to support TD identification from source code comments. The results pointed that the dimensions (e.g. Adjectives, Adverbs, Verbs, Nouns, and Tags) considered by the model are used by developers when writing comments and that CVM-TD can be effectively used to support TD identification activities.

These promising initial outcomes motivated us to further evaluate CVM-TD with other projects. Therefore, in this paper we extend the research of (Farias *et al.,* 2015) with an additional study to analyze the use of CVM-TD and the contextualized vocabulary with the purpose of characterizing its overall accuracy when classifying candidate comments and factors that influence the analysis of the comments to support the identification of TD in terms of accuracy.

We address this research goal by conducting a controlled experiment to investigate the overall CVM-TD accuracy and the influence of the English skills and developer experience factors. We analyzed the factors against the accuracy by observing the agreement between each participant and an oracle elaborated by the researchers. We compared the accuracy values for the different factors using statistical tests.

We also analyzed the agreement among the participants. These aspects are decisive to understand and validate the model and the contextualized vocabulary. Our findings indicate that CVM-TD provided promising results considering the accuracy values. The accuracy values of the participants with good reading skills were better that the values of the participants with medium/poor reading skills. We could not conclude that the experience level affects the accuracy when identifying TD items through comment analysis. We also observed that many comments had high agreement, almost 60% of comments filtered by terms that belong to the vocabulary (candidate comments) proposed in (Farias *et al.,* 2015) were identified as good indicators of TD.

The remainder of this paper is organized as follows. Section 2 presents relevant literature reporting on technical debt identification approaches and the use of comments in source code. Section 3 describes the planning of the controlled experiment. Section 4 presents its operation. The results are presented in Section 5. Next, we have a discussion section. Finally, Section 7 concludes the paper.

2 BACKGROUND

2.1 Code Comments Mining

Comments are an important software artifact which may help to understand software features (Storey *et al.,* 2008). Code comments have been used as data source in some research (Storey *et al.,* 2008) (Maalej and Happel, 2010).

In (Maalej and Happel, 2010), the authors analyzed the purpose of work descriptions and code comments aiming to discuss how automated tools can support developers in creating them.

(Storey *et al.,* 2008) analyzed how developers deal with software maintenance tasks by conducting an empirical study investigating how comments may improve processes and tools that are used for managing these tasks.

In fact, comments have been used to describe issues that may require future work, emerging problems and decisions taken about those problems (Maalej and Happel, 2010). These descriptions facilitate human readability and provide additional information that summarizes the developer context (Farias *et al.,* 2015).

2.2 Using Code Comments to Identify TD

More recently, code comments have been explored with the purpose of identifying TD (Potdar and Shihab, 2014) (Maldonado and Shihab, 2015) (Farias *et al.,* 2015).

(Potdar and Shihab, 2014) analyzed code comments to identify text patterns and TD items. They read more than 101K code comments. Their findings show that 2.4 - 31.0% of the files in a project contain self-admitted TD. In addition, the most used text patterns were: (i) "is there a problem" with 36 instances, (ii) "hack" with 17 instances, and (iii) "fixme" with 761 instances.

In another TD identification approach, (Maldonado and Shihab, 2015) evolved the work of (Potdar and Shihab, 2014) proposing four simple filtering heuristics to eliminate comments that are not likely to contain technical debt. For that, they read 33K code comments from source code of five open source projects (Apache Ant, Apache Jmeter, ArgoUML, Columba, and JFreeChart). Their findings showed that self-admitted technical debt can be classified into five main types: design debt, defect debt, documentation debt, requirement debt, and test debt. According to the authors, the most common type of self-admitted TD is design debt (between 42% and 84% of the classified comments).

In the same sense, Farias *et al.* proposed the CVM-TD (Farias *et al.,* 2015). CVM-TD is a contextualized structure of terms that focuses on using word classes and code tags to provide a TD vocabulary, aiming to support the detection of different types of debt through code comment analysis. In order to evaluate the model and quickly analyze developers' comments embedded in source code, the *eXcomment* tool was developed. This tool extracts and filters candidate comments from source code using the contextualized vocabulary provided by CVM-TD.

This research provided preliminary indication that CVM-TD and its contextualized vocabulary can be effectively used to support TD identification (the whole vocabulary can be found at https://goo.gl/TH2ec5). However, the factors that may affect its accurate usage are still unknown. In this work, we focused on characterizing CVM-TD's accuracy and some of these factors.

3 STUDY PLANNING

3.1 Goal of Study and Research Questions

This study aims at investigating the following goal: "**Analyze** the use of CVM-TD **with the purpose of** characterizing its overall accuracy and factors affecting the identification of TD through code comment analysis, **with respect to** accuracy when identifying TD items **from the point of view of** the researcher **in the context of** software engineering master students with professional experience analyzing candidate code comments of large software projects"**.** More specifically, we investigated four Research Questions (RQ). The description of these RQs follows.

RQ1: *Do the English reading skills of the participant affect the accuracy when identifying TD through code comment analysis?*

Considering that non-native English speakers are frequently unaware of the most common terms used to define specific parts of code in English (Lemos *et al.,* 2014), this question aims to investigate whether a different familiarity with the English language could impact the identification of TD through code comment analysis. In order to analyze this variable, we split the participants into levels of "good English reading skills" and "medium/poor English reading skills". This question is important to help us to understand the factors that may influence the analysis of comments to identify TD.

RQ2: *Does the experience of the participant affect the accuracy when identifying TD through code comment analysis?*

Experience is an important contextual aspect in the software engineering area (Host *et al.*, 2005). Recent research has studied the impact of experience on software engineering experiments (Salman, 2015). Some works have found evidence that experience affects the identification of code smells, and that some code smells are better detected by experienced participants rather than by automatic means (Santos *et al.*, 2014). Considering this context, this question aims to discuss the impact of the participants' experience on the identification of TD through code comment analysis. In order to analyze the variable, we classified the participants into three levels considering their experience with software development: i) high experience, ii) medium experience, and iii) low experience. This question is also important to help us to understand the factors that may influence the analysis of comments to identify TD.

RQ3: *Do participants agree with each other on the choice of comments filtered by CVM-TD that may indicate a TD item?*

With this question, we intend to investigate the contribution of CVM-TD in the TD identification process and how many and what comments had high level of agreement. That is, what comments point out to a TD item. This will also allow us to analyze the agreement among the participants about the candidate

comments that indicate a TD item. We conjecture that high agreement on the choice of comments filtered by CVM-TD evidences its relevance as a support tool on the TD identification.

RQ4: *Does CVM-TD help researchers on select candidate comments that point to technical debt items?*

With this question, we intend to investigate if the contextualized vocabulary provided by CVM-TD points to candidate comments that are considered indicators of technical debt by researchers. This will also allow us to investigate the contribution of CVM-TD to support the TD identification process.

3.2 Participants

The participants of the study were selected using convenience sampling (Shull and Singer, 2008). Our sample consists of 21 software engineering master students at the Federal University of Sergipe (Sergipe-Brazil) and 15 software engineering master students at the Salvador University (Bahia-Brazil). We conducted the experiment in the context of the Empirical Software Engineering course.

In order to classify the profile of the participants and their experience in the software development process, a characterization form was filled by each participant before the experiment. The questions were about professional experiences, English reading skills, and specific technical knowledge such as refactoring and programming languages. The result of the questionnaire showed that participants had a heterogeneous experience level, but all had some type of experience on software projects.

The participants were classified into three experience levels (high, medium and low) regarding the experience variable and the classification proposed by (Host *et al.,* 2005), which is presented in Table 1. We discarded the category E1 because there were not any undergraduate students as participants. We considered low experience for participants related to the categories E2 and E3. The participants related to the category E4 were considered as having medium experience, and, finally, we considered the participants related to category E5 as having high experience.

When considering the English reading skills, the participants were classified into two levels (good and medium/poor). We had 4 participants with poor English reading skills, and 21 participants with medium. Despite these participants have been selected as medium/poor English, they may understand short sentences like code comments in English. Table 2 shows the characterization of the participants.

Table 1: Classification of the experiences of participants.

Category	Description	Experience levels
E1	Undergraduate student with less than 3 months of recent industrial experience	--
E2	Graduate student with less than 3 months of recent industrial experience	Low
E3	Academic with less than 3 months of recent industrial experience	Low
E4	Any person with recent industrial experience between 3 months and 2 years	Medium
E5	Any person with recent industrial experience for more than 2 years	High

The participants were split into three groups. Each group had 12 participants with approximately the same levels of experience. This strategy provides a balanced experimental design. The design involved each group of participants working on a different set of comments (experimental object), and permits us to use statistical test to study the effect of the investigated variables. We adopted this plan in order to avoid an excessive number of comments to be analyzed by each participant.

3.3 Instrumentation

3.3.1 Forms

The experimental package is available at https://goo.gl/DdomGk. We used slides for the training and four forms to perform the experiment:

Consent Form: the participants authorize their participation in the experiment and indicate to know the nature of the procedures which they have to follow.

Characterization Form: contains questions to gather information about professional experiences, English reading skills, and specific technical knowledge of participants.

Data Collect Form: contains a list of source code comments. During the experiment, the participants were asked to indicate, for each comment, if it points to a TD item.

Table 2: Distribution of the participants.

Group	Participants by experience level			Participants by English reading level	
	High	Med	Low	Good	Med/Poor
G1 (12)	4	3	5	1	11
G2 (12)	3	5	4	5	7
G3 (12)	4	5	3	5	7
Total (36)	**11**	**13**	**12**	**11**	**25**

Feedback Form: in this form, the participants may write their impression on the experiment. We also asked the participants to classify the training and the level of difficulty in performing the study tasks.

3.3.2 Software Artifact and Candidate Comments

We gathered and filtered comments from a large and well-known open source software (ArgoUML). The project is written in Java with 1,846 files. In choosing this project, we considered the following criteria: being long-lived (more than 10 years), having a satisfactory number of comments (more than 2,000 useful comments).

To be able to extract the candidate comments from the software that may indicate a TD item, we used *eXcomment*. We were only interested in comments that have been intentionally written by developers (Farias *et al.,* 2015).

Once the comments were extracted, we filtered the comments by using terms that belong to the vocabulary presented in (Farias *et al.,* 2015). A comment is returned when it has at least one keyword or expression found in the vocabulary. We will call these comments 'candidate comments'. At the end, the tool returned 353 comments, which were listed in the data collect form in the same order in which they are in the code. This is important because comments that are close to each other can have some kind of relationship.

3.4 Analysis Procedure

We considered three perspectives to analyze accuracy:

(i) Agreement between Each Participant and the Oracle: In order to investigate RQ1 and RQ2, we adopted the accuracy measure, which is the proportion of true results (the comments chosen in agreement between each participant and the oracle) and the total number of cases examined (see Equation 1).

$$accurary = (num\ TP + num\ TN) / (num\ TP + num\ FP + num\ TN + num\ FN) \quad (1)$$

TP represents the case where the participant and the oracle agree on a TD comment (comment that points to a TD item). FP represents the case where the participant disagrees with the oracle with respect to the selected TD comment. TN occurs when the participant and the oracle agree on a comment that does not report a TD item. Finally, a FN happens when the participant does not mark a TD comment in disagreement with the oracle.

The definition of the oracle, which represents an important aspect of this analysis process, was performed prior to carrying out the experiment. We relied on the presence of three specialists in TD. Two of the specialists did, in separate, the indication of the comments that could point out to a TD item. After, the third specialist did a consensus process for the set of the chosen comments. All this process took one week.

(ii) Agreement among the Participants: To analyze RQ3, we adopted the Finn coefficient (Finn, 1970). The Finn coefficient is used to measure the level of agreement among participants. In order to make the comparison of agreement values, we adopted classification levels, as defined by (Landis and Koch, 1977), and recently used by (Santos *et al.,* 2014): slight, for values between 0.00 and 0.20; fair (between 0.21 and 0.40); moderate (between 0.41 and 0.60); substantial (between 0.61 and 0.80); and almost perfect (between 0.81 and 1.00) agreement.

(iii) TD Comments Selected by Oracle: To analyze RQ4, we investigated the candidate comments that point to TD items selected by the oracle.

3.5 Pilot Study

Before the experiment, we carried out a pilot study with a computer science PhD student with professional experience. The pilot took 2 hour and was carried out in a Lab at the Federal University of Bahia (Bahia-Brazil). We performed the training at first hour, and next the participant performed the experimental task described in the next section. The participant analyzed 83 comments and selected 52 as TD comments.

The pilot was used to better understand the procedure of the study. It helped us to evaluate the use of the data collection form, the necessary time to accomplish the task and, mainly, the number of comments used by each group. Thus, the pilot study was useful to calibrate the time and number of comments analyzed.

4 OPERATION

The experiment was conducted in a classroom at Federal University of Sergipe, and at the Salvador University, following the same procedure.

The operation of the experiment was divided into different sessions. A week prior to the experiment, the participants filled the consent and characterization form. The training and experiment itself were performed at the same day. For training purposes, we performed a presentation in the first part of the class. The presentation covered the TD concepts and context, as well as the TD indicators (Alves *et al.,* 2014) and how to perform a qualitative analysis on the code comments. This training took one hour.

After that, a break was taken. Next, each participant analyzed the set of candidate comments, extracted from ArgoUML, in the same room where the training was provided. They filled the data collection form pointing out the initial and end time of the task. For each candidate comment listed in the form, the participants chose "Yes" or "No", and their level of confidence on their answer. They used an ordinal scale of one to four to represent the confidence. Besides, for each comment marked as yes, they should highlight the piece of text that was decisive for giving this answer.

The participants were asked to not discuss their answers with others. When they finished, they filled the feedback questionnaire. A total of three hours were planned for the experiment training and execution, but the participants did not use all of the available time.

4.1 Deviations from the Plan

We did not include the data points from participants who did not complete all the experimental sessions in our analysis, since we needed all the information (characterization, data collection, and feedback). Thus, we eliminated 4 participants.

Table 3 presents the final distribution of the participants. The value in parentheses indicates the final number of participants in each group. In each of the groups G1 and G3, a participant was excluded because of not filling the value of confidence. In group G2, a participant was excluded because he did not analyze all comments and another was excluded because did not mark the text in the TD comments.

5 RESULTS

In this section, we present the results for each RQ.

RQ1: *Do the English reading skills of the participant affect the accuracy when identifying TD through code comment analysis*?

In order to investigate the impact of the English reading level skills on the TD identification process, we calculated the accuracy values for each participant with respect to the oracle. Figure 1 shows a box plot illustrating the accuracy distribution. It is possible to note that the participants with good English reading skills had the lowest dispersion. It indicates that they are more consistent in the identification of TD comments than the participants with medium/poor English reading skills. Moreover, the accuracy values of the participants with good reading skills are higher than the values of the participants with medium/poor reading skills. However, the median accuracy of the participants with medium/poor reading skills is 0.65. This means that the participants with this profile were able to identify comments that were pointed out as an indicator of a TD item by the oracle.

We also performed a hypothesis test to reinforce the analysis of this variable. To do this, we defined the following null hypothesis:

H0: *The English reading skills of the participant do not affect the accuracy with respect to the agreement with the oracle.*

We ran a normality test, *Shapiro-Wilk*, and identified that the distribution was normal. After that, we ran the *t-test*, a parametric test, to evaluate our hypotheses. We used a typical confidence level of 95% ($\alpha = 0.05$). As shown in Table 4, the p-value calculated ($p=0.02342$) is lower than the α value. Consequently, we may reject the null hypothesis (*H0*).

We also evaluated our results in terms of magnitude, testing the effect size measure. We calculate *Cohen's d* (Cohen, 1988) in order to interpret the size of the difference between the distribution of the groups. We used the classification presented by Cohen (Cohen, 1988): 0 to 0.1: No

Table 3: Final distribution of the participants among groups.

Group	Participants by experience level			Participants by English level	
	High	Medium	Low	Good	Med/ Poor
G1 (11)	4	3	4	1	10
G2 (10)	2	5	3	5	5
G3 (11)	3	5	3	5	6
Total (32)	**9**	**13**	**10**	**11**	**21**

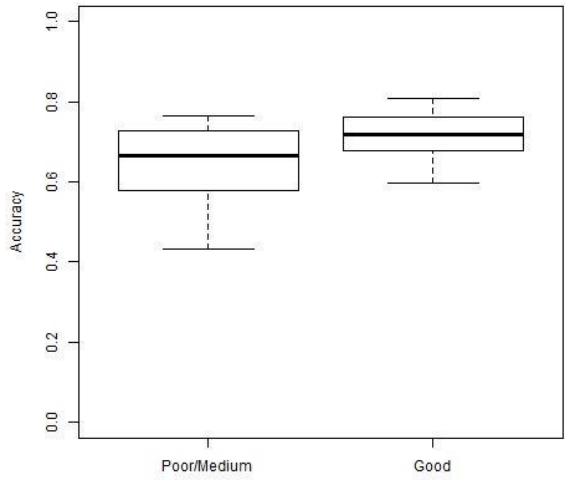


Figure 1: Accuracy value by English reading skills.

Effect; .2 to .4: Small Effect; .5 to .7: Intermediate Effect; .8 and higher: Large Effect.

The magnitude of the result (d = 0.814) also confirmed that there are a difference (Large Effect) on the accuracy values with respect to both groups. This evidence reinforces our hypothesis and shows that the results were statistically significant.

In addition, we analyzed the feedback form and we highlighted the main notes at the following (translated to English): (i) *I had some difficulties to understand and decide about complex comments*; (ii) *I had the feeling that I needed to know the software context better*; (iii) *I believe some tips on English comments could help us to interpret the complex comments*. This data is aligned with our finding that indicates that English reading skills may affect the task of analyzing code comments to identify TD in software projects.

Table 4: Hypothesis test for analysis of English reading.

	Shapiro-Wilk (Normality Test)		Parametric Test
	Good	Medium/Poor	t-test
p-value	0.9505	0.9505	0.02342

RQ2: *Does the experience of participant impact the accuracy when identifying TD through code comment analysis?*

In order to investigate the impact of the experience level on the TD identification process, we calculated the accuracy values for each participant with respect to the oracle. We show the accuracy distribution by experience level of the participants in Figure 2. From this figure, it is possible to note that the box plots have almost the same level of accuracy regarding high, medium and low experience. Considering the median values, the values are very similar. Participants with high and low experience have the same median value (0.66), whereas the median of participants with medium experience is moderately higher (0.71).

We also calculated the variation coefficient. This coefficient measures the variability in each level – that is, how many in a group is near the median. We found the coefficients of 12.91%, 13.17%, and 13.96%, for high, medium and low experience, respectively. According to the distribution presented by (Snedecor and Cochran, 1967), the coefficients are low, showing that the levels of experience have homogeneous values of accuracy.

Finally, we performed a hypothesis test to analyze the experience variable. We defined the following null hypothesis:

H0: *The experiences of the participants do not affect his or her accuracy with respect to the agreement with the oracle.*

After testing normality, we ran Anova, a parametric test to evaluate more than two treatments. The p-value calculated (p= *0.904*) is bigger than α value. In this sense, we do not have evidences to reject the null hypothesis (*H0*).

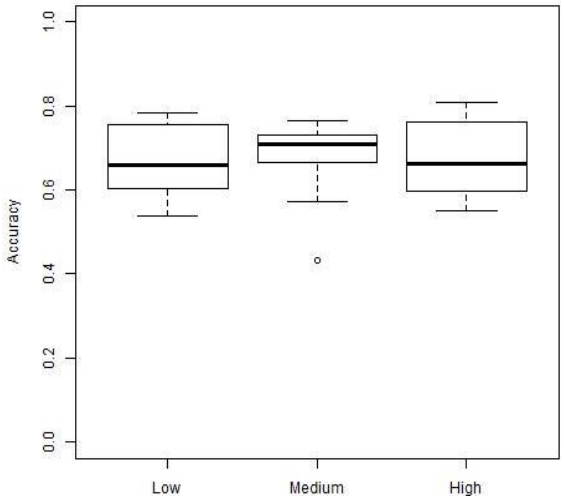


Figure 2: Accuracy by participants' experience.

From the analysis, we consider that the experience level did not impact the distribution of the accuracy values, i.e., when using CVM-TD, experienced and non-experienced participants show the same accuracy when identifying comments that point out to TD items. A possible interpretation of this result is that CVM-TD can be used by non-experienced participants.

RQ3: *Do participants agree with each other on the choice of comments filtered by CVM-TD that may*

indicate a TD item?

Our analysis considered the number of comments per percentage of participants that chose the comment. Figure 3 shows the percentages in the *X-axis,* and the number of comments in each interval in *Y-axis*. The percentage values are the proportion of "number of participants who choose the comment" and the "number of participants in the experiment group". For example, a comment from group G1 (the G1 has 11 participants) that was chosen by 10 participants has ratio = 0.91 (that is, 10/11).

It is possible to note that some comments were chosen as good indicator of TD by all or almost all participants, which means that these comments had high level of agreement and CVM-TD filtered comments that may really point out to TD item. Almost 40 comments have ratio intervals between 1 and 0.90. Some examples of such comments are: "*NOTE: This is temporary and will go away in a "future" release (ratio = 1)"; "// FIXME: this could be a problem...(ratio = 1)", "TODO: Replace the next deprecated call (ratio = 0.90)". "TODO: This functionality needs to be moved someplace useful…(ratio = 0.90)".* The whole set of these comments is available at https://goo.gl/fSaMj9.

On the other hand, considering the agreement among all participants identifying TD comments, we found a low coefficient. We conducted the Finn test to analyze the agreement in each group, considering all comments. Table 5 presents the agreement coefficient values. The level of agreement was 'slight' and 'fair' according to (Landis and Koch, 1977) classification.

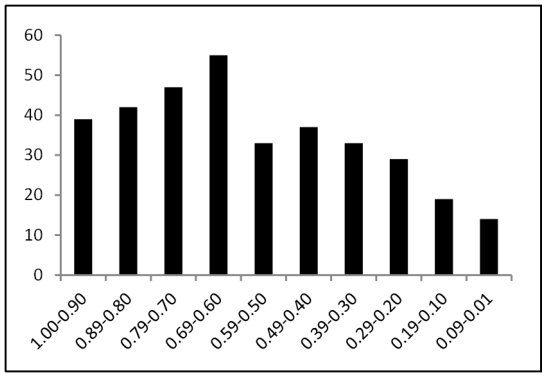


Figure 3: Agreement among TD comments.

Table 5: Finn agreement test.

	Finn	p-value	Classification levels
Group 1	0.151	3.23e-05	Slight
Group 2	0.188	5.74e-07	Slight
Group 3	0.265	8.34e-12	Fair

RQ4: *Does CVM-TD help researchers on select candidate comments that point to technical debt items?*

We analyzed the candidate comments identified by the oracle as TD comments. Table 6 shows the number of comments identified by the oracle. We observed that almost 60% of comments filtered by terms that belong to the vocabulary (candidate comments) proposed in (Farias *et al.,* 2015) were identified as good indicators of TD by the oracle.

6 DISCUSSION

Our results suggest that the English reading level of the participants may impact the identification of TD through comment analysis. Participants with good English reading skills had accuracy values better than participants who have medium/poor English reading skills. On the other hand, participants with poor/medium English profile were able to identify a good amount of TD comments filtered by the contextualized vocabulary.

We also observed in the feedback analysis that some participants had difficulties to understand and interpret complex comments, and tips might help them with this task. We conjecture that some tips may support participants to make decision on the TD identification process. For instance, highlight the TD terms or patterns of comment from the contextualized vocabulary into the comments.

Considering the impact of experience on TD identification, we could not conclude that the experience level affects the accuracy. This can indicate that comments selected by the vocabulary may be understood by an experienced or non-experienced observer. This reinforces the idea that the TD metaphor aids discussion by providing a familiar framework and vocabulary that may be easily understood (Spínola *et al.,* 2013)(Kruchten *et al.*, 2012).

Table 6: TD comments identified by the oracle.

	Group 1	Group 2	Group 3
Number of candidate comments	123	124	106
Number of TD comments	68 (55.28%)	83 (66.94%)	58 (54.72%)

Considering the agreement among participants identifying TD comments, the results revealed some comments pointed out as good indicator of TD, with

high level of agreement. It may evidence the contribution of CVM-TD as a support tool on the TD identification. However, in general, the level of agreement between participants was considered low. We believe that this occurred due to the large amount of comments to be analyzed, and the amount of comments selected by the contextualized vocabulary that does not indicate a TD item. In this way, the level of agreement might rise whether the vocabulary is more accurate.

The last aspect we analyzed was the contribution of the CVM-TD to support TD identification. We noted that a high number of comments filtered by the CVM-TD was considered as TD item. These results provide preliminary indications that CVM-TD and the contextualized vocabulary can be considered an important support tool to identify TD item through code comments analysis. Different from code metrics-based tools, code comments analysis allow us to consider human factors in order to explore developers' point of view and complement the TD identification with more contextual and qualitative data. Both approaches may contribute with each other to make the automated tools more efficient.

6.1 Threats to Validity

We followed the checklist provided by (Wohlin and Runeson, 2000) to discuss the relevant threats to this controlled experiment.

6.1.1 Construct Validity

To minimize the mono-method bias, we used an accuracy and agreement test to provide an indication of the TD identification through comment analysis. The researchers that composed the oracle were selected by authors of this study. In order to mitigate the biased judgment on the oracle, its definition was performed by three different researchers with knowledge in TD. Two of them selected the TD comments and the third researcher did a consensus to decrease the bias. Finally, to reduce social threats due to evaluation apprehension, participants were not evaluated.

6.1.2 Internal Validity

The first internal threat we have to consider is subject selection, since we have chosen all participants through a convenience sample. We minimized this threat organizing the participants in different treatment groups divided by experience level.

Another threat is that participants might be affected negatively by boredom and tiredness. In order to mitigate this threat, we performed a pilot study to calibrate the time and amount of comments to be analyzed. To avoid the communication among participants, two researchers observed the operation of the experiment at all times. A further validity threat is the instrumentation, which is the effect caused by artifacts used for the experiment. Each group had a specific set of comments, but all participants used the same data collection form format. In order to investigate the impact of this threat in our results, we analyzed the average accuracy in each group. Group G1 has average value equal to 0.65. For group G2, the average value is equal to 0.66, and group G3 is equal to 0.69. From these data, it is possible to note that groups have almost the same level of average accuracy. It means that this threat did not affect the results.

6.1.3 External Validity

This threat relates to the generalization of the findings and their applicability to industrial practices. This threat is always present in experiments with students as participants. Our selected samples contained participants with different levels of experience. All participants have some professional experience in the software development process. It is an important aspect in mitigating the threat. A further threat is the usage of software that may not be representative for industrial practice. We used software adopted in the practice of software development as an experimental object in order to mitigate the threat.

6.1.4 Conclusion Validity

To avoid the violation of assumptions, we used normality test, Shapiro-Wilk, and a parametric test, the t-test, for data analysis. To reduce the impact of reliability of treatment implementation, we followed the same experimental setup on both cases.

7 CONCLUSION AND FUTURE WORK

In this paper, we performed a controlled experiment in order to evaluate the CVM-TD aiming to characterizing its overall accuracy and factors that may affect the identification of TD through code comment analysis. Our results indicate that: (i) English reading skills affect the participants' accuracy; (ii) we could not conclude that the

experience level impacts on understanding of comments to support the TD identification; (iii) concerning the agreement among participants, although we found low agreement coefficients between participants, some comments have been indicated with a high level of agreement; (iv) CVM-TD provided promising results concerning to the identification of comments as good indicator of TD by participants. Almost 60% of the candidate comments filtered by CVM-TD were identified as actual TD indicators by oracle.

The results motivate us to continue exploring code comments in the context of the TD identification process in order to improve CVM-TD and the *eXcomment*. Future works include to: (i) develop some feature in *eXcomment* associated with the CVM-TD to support the interpretation of comments, such as "usage of weights and color scale to indicate the comments with more importance in TD context, and highlight the TD terms or patterns of comment into the comments", and (ii) evaluate the use of CVM-TD in projects in the industry.

ACKNOWLEDGEMENTS

This work was partially supported by CNPq Universal 2014 grant 458261/2014-9. The authors also would like to thank Methanias Colaço for his support in the execution step of the experiment.

REFERENCES

Alves, N.S.R. et al., 2016. Identification and Management of Technical Debt: A Systematic Mapping Study. *Information and Software Technology*, pp.100–121.

Alves, N.S.R. et al., 2014. Towards an Ontology of Terms on Technical Debt. *6th MTD*. pp. 1–7.

Wohlin, C and Runeson, M.H., 2000. Experimentation in Software Engineering: an introduction, Kluwer *Academic Publishers Norwell.*

Ernst, N.A. et al., 2015. Measure It ? Manage It ? Ignore It ? Software Practitioners and Technical Debt. *10th Joint Meeting on Found. of Soft. Engineering. ACM.*

Farias, M. et al., 2015. A Contextualized Vocabulary Model for Identifying Technical Debt on Code Comments. 7*th MTD*. pp. 25–32.

Finn, R.H., 1970. A Note on Estimating the Reliability of Categorical Data. *Educational and Psychological Measurement*, pp.71–76.

Guo, Y. *et al.*, 2014. Exploring the costs of technical debt management – a case study. *ESE*, 1, pp.1–24.

Host, M., Wohlin, C. and Thelin, T., 2005. Experimental context classification: incentives and experience of subjects. *27th ICSE*, pp.470–478.

Izurieta, C. et al., 2012. Organizing the technical debt landscape. 2012 *3rd MTD,* pp.23–26.

Cohen, J. 1988. Statistical power analysis for the behavioral sciences. 2 edition. L. Erlbaum, ed.,

Kruchten, P. *et al.*, I., 2012. Technical debt: From metaphor to theory and practice. *IEEE,* pp.18–21.

Landis, J.R. and Koch, G.G., 1977. The measurement of observer agreement for categorical data. *Biometrics*, pp.159–174.

Lemos, O. a L. et al., 2014. Thesaurus-Based Automatic Query Expansion for Interface-Driven Code Search Categories and Subject Descriptors, pp.212–221.

Li, Z. et al., 2014. A systematic mapping study on technical debt. *Journal of Syst. Soft.* 101, pp.193–220.

Maalej, W. and Happel, H.-J., 2010. Can development work describe itself? *7th MSR*, pp.191–200.

Maldonado, E.S. and Shihab, E., 2015. Detecting and Quantifying Different Types of Self-Admitted Technical Debt. *In 7th MTD*. pp. 9–15.

Mendes, T. et al., 2015. VisMinerTD - An Open Source Tool to Support the Monitoring of the Technical Debt Evolution using Software Visualization. *17th ICEIS*.

Potdar, A. and Shihab, E., 2014. An Exploratory Study on Self-Admitted Technical Debt. *ICSME,* pp. 91–100.

Salman, I., 2015. Are Students Representatives of Professionals in Software Engineering Experiments? *37th ICSE*. IEEE Press, 2015.

Santos, J.A.M., et al., 2014. The problem of conceptualization in god class detection : agreement , strategies and decision drivers. J*ournal of Software Engineering Research and Development*, (2), pp.1–33.

Shull, F., Singer, J. and Sjoberg, D., 2008. Guide to Advanced Empirical Software Engineering, *Springer*.

Snedecor, G.W. and Cochran, W.G., 1967. Statistical Methods. Ames.

Spínola, R. et al., 2013. Investigating Technical Debt Folklore. *5th MTD*, pp.1–7.

Storey, M. et al., 2008. TODO or To Bug : Exploring How Task Annotations Play a Role in the Work Practices of Software Developers. *ICSE*. pp. 251–260.

Zazworka, N. et al., 2013. A case study on effectively identifying technical debt. *17thEASE*. ACM, pp.42–47.

JCL: A High Performance Computing Java Middleware

André Luís Barroso Almeida[1,2], Saul Emanuel Delabrida Silva[1], Antonio C. Nazare Jr.[3] and Joubert de Castro Lima[1]

[1]*DECOM, Universidade Federal de Ouro Preto, Ouro Preto, Minas Gerais, Brazil*

[2]*CODAAUT, Instituto Federal de Minas Gerais, Ouro Preto, Minas Gerais, Brazil*

[3]*DCC, Universidade Federal de Minas Gerais, Belo Horizonte, Minas Gerais, Brazil*

andre.almeida@ifmg.edu.br, {saul.delabrida, joubert}@iceb.ufop.br, antonio.nazare@dcc.ufmg.br

Keywords: Big Data, Internet of Things, Middleware, Reflective Computing, High Performance Computing, Distributed Shared Memory, Remote Method Invocation.

Abstract: Java Cá&Lá or just JCL is a distributed shared memory reflective lightweight middleware for Java developers whose main goals are: *i*) provide a simple deployment strategy, where automatic code registration occurs, *ii*) support a collaborative multi-developer cluster environment where applications can interact without explicit dependencies, *iii*) execute existing sequential Java code over both multi-core machines and cluster of multi-core machines without refactorings, enabling the separation of business logic from distribution issues in the development process, *iv*) provide a multi-core/multi-computer portable code. This paper describes JCL's features and architecture; compares and contrasts JCL to other Java based middleware systems, and reports performance measurements of JCL applications.

1 INTRODUCTION

We live in a world where large amounts of data are stored and processed every day (Han et al., 2011). Despite the significant increase in the performance of today's computers, there are still big problems that are intractable by sequential computing approaches (Kaminsky, 2015). Big data (Brynjolfsson, 2012), Internet of Things (IoT) (Perera et al., 2014) and elastic cloud services (Zhang et al., 2010) are results of this new decentralized, dynamic and communication-intensive society. Many fundamental services and sectors such as electric power supply, scientific/technological research, security, entertainment, financial, telecommunications, weather forecast and many others, use solutions that require high processing power. For instance, "Walmart handles more than a million customer transactions each hour and it estimated that the transaction database contains more than 2.5 petabytes of data" (Troester, 2012). Thus, these solutions are executed over parallel and distributed computer architectures. In this context, a new demand for both computer architectures and applications to handle such big problems has emerged.

High Performance Computing (HPC) is based on the concurrence principle, so high speedups are achievable, but the development process becomes complex when concurrence is introduced. Therefore, middleware systems and frameworks are designed to help reducing the complexity of such development.

The use of middleware as a software layer on top of an operating system became usual in last years in order to organize a computer cluster or grid (Tanenbaum and Van Steen, 2007). The challenging issue is how to provide sufficient support and general high-level mechanisms using middleware for rapid development of distributed and parallel applications. Furthermore, the middleware systems found in the literature have no information of their use on different computational platforms. For instance, there is no evidence that a set of embedded devices are able to work with a cloud platform using the same middleware. To developers, the system integration is not transparent and depends of different skills. In Perera et al. (2014), the authors mention the existence of six machine classes in IoT and they advocate that middleware systems should run over all of them (Perera et al., 2014).

Middleware systems can be adopted for general purposes, such as Message Passing Interface (MPI) (Forum, 1994), Java Remote Method Invocation (RMI) (Pitt and McNiff, 2001), Hazelcast (Veentjer, 2013), JBoss (Watson et al., 2005) and many others, but they can also be designed for a specific

Almeida, A., Silva, S., Jr., A. and Lima, J.
JCL: A High Performance Computing Java Middleware.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 379-390
ISBN: 978-989-758-187-8

purpose, like gaming, mobile computing and real-time computing, for instance (Murphy et al., 2006; Gokhale et al., 2008; Tariq et al., 2014). They support a programming model based on shared memory, message passing or event based (Ghosh, 2014). Among various programing languages for middleware systems, the interest in Java for HPC is rising (Taboada et al., 2013). This interest is based on many features, such built-in networking, multithreading support, platform independence, portability, type safety, security, extensive API and wide community of developers (Taboada et al., 2009).

Besides the computational performance, the issues presented below must be considered in the design and development of a modern middleware.

Refactorings: Usually, middleware systems introduce some dependencies to HPC applications, thus, end-users need to follow some standards specified by them, since methods and global variables must be distributed over a data network. Consequently, an ordinary Java or C++ object [1] must implement several middleware classes or statements to become distributed. There are many middleware examples with such dependencies, including standards and market leaders like Java RMI (Pitt and McNiff, 2001), JBoss (Watson et al., 2005) and Mapreduce based solutions (Hindman et al., 2011; Zaharia et al., 2010). As a consequence of these dependencies, two problems emerge: *i*) the end-user cannot separate business logic from distribution issues during the development process and; *ii*) existing and well tested sequential applications cannot be executed over HPC architectures without refactorings. A zero-dependency middleware is unrealistic, but a middleware with few adaptations is fundamental to achieve low coupling with the existing code.

Deployment: Deployment can be a time consuming task in large clusters, i.e. any live update of an application module or class often interrupts the execution of all services running in the cluster. Some middleware systems adopt third-party solutions to distribute and update modules in a cluster (Henning and Spruiell, 2006; Nester et al., 1999; Veentjer, 2013; Pitt and McNiff, 2001), but sometimes updating during application runtime and without stoppings is a requirement. This way, middleware systems capable of deploying a distributed application transparently, as well as updating its modules during runtime and programmatically, are very useful to reduce maintenance costs caused by several unnecessary interruptions.

Collaboration: Cloud computing introduces opportunities, since it allows collaborative development or development as a service in cloud stack. A middleware providing a multi-developer environment where applications can access methods and user typed objects from each other without explicit references is fundamental to introduce development as a service or just to transform a cluster into a collaborative development environment.

Portable Code: Portable multi-core/multi-computer code is an important aspect to consider during development process, since in many institutions, such as research ones, there can be huge multi-core machines and several clusters of ordinary PCs to solve a couple of problems. This way, code portability is very useful to test algorithms and data structures in different computer architectures without refactorings. A second justification for offering at least two releases in a middleware is that clusters are nowadays multi-core, so middleware systems must implement shared memory architectural designs in conjunction with distributed ones.

The main goal of this work is to introduce a new middleware that fill the issues previously shown, precisely:*i*) a simple deployment strategy and capacity to update internal modules during runtime; *ii*) a collaborative multi-developer environment; *iii*) a service to execute existing sequential Java code over both multi-core machines and cluster of multi-core machines without refactorings; *iv*) multi-core/multi-computer portable code. This middleware, called Java Cá&Lá[2] or just JCL, is a tool for develop HPC applications. In this paper, the three main components of our architecture are presented and evaluated. Is not the focus of this work to be a "how to" guide, although some source codes are shown in order to clarify the reader's understanding. The main contributions of JCL are:

i) A middleware that gathers several features presented separately in the last decades of middleware literature, enabling building distributed applications with few portable instructions to clusters made from different platforms;

ii) A comparative study of market leaders and well established middleware standards for Java community. This paper emphasizes the importance of several features and how JCL and its counterparts fulfill them;

iii) A scalable middleware over multi-core and multi-computer architectures;

iv) A feasible middleware alternative to fast prototype portable Java HPC applications.

This work is organized as follows. Section 2 discusses works that are similar to the proposed middle-

[1] "An object is a self-contained entity consisting of data and procedures to manipulate data" (Egan, 2005)

[2] Java Cá&Lá is available for download at http://www.joubertlima.com.br

ware, pointing out their benefits and limitations. Section 3 details the JCL middleware, presenting its architecture and features. Section 4 describes a user case application. Section 5 presents our experimental evaluation and discusses the results. Finally, in Section 6, we conclude our work and point out future improvements of JCL.

2 RELATED WORK

In this section, we describe the most promising middleware systems in various stages of development. We evaluated each work in terms of: *i*) requirement for low/medium/high refactorings, *ii*) automatic or manual deployment, *iii*) support for collaborative development, *iv*) implementation of both multi-core and multi-computer portable code. Other analyses were made to verify if the middleware is discontinued, and if it is fault tolerant in terms of storage and processing. Academic and commercial solutions are put together and their limitations/improvements are highlighted in Table 1. Middleware systems that present high similarities with JCL are described in detail in this section. The remaining related work is described just in Table 1.

Infinispan by JBoss/RedHat (Team, 2015) is a popular open source distributed in-memory key/value data store (Di Sanzo et al., 2014) which enables two ways to access the cluster: *i)* the first way enables an API avaliable in a Java library ; *ii)* the second way enables several protocols, such as HotRod, REST, Memcached and WebSockets, making *Infinispan* a language independent solution. Besides storage services, the middleware can execute tasks remotely and asynchronously, but end-users must implement Runnable or Callable interfaces. Furthermore, it is necessary to register these tasks at Java virtual machine (JVM) classpath of each cluster node, so *Infinispan* does not have the dynamic class loading feature.

Java Parallel Processing Framework *JPPF* is an open source grid computing framework based on pure Java language (Xiong et al., 2010) which simplifies the process of parallelizing applications that demand high processing, allowing end-users to focus on their core software development (Cohen, 2015). It implements the dynamic class loading feature in cluster nodes, but it does not support collaborative development, i.e. methods cannot be shared among different *JPPF* applications, producing many services over a cloud infrastructure, for instance. JPPF does not implement shared memory services just execute methods.

Hazelcast (Veentjer, 2013) is a promising middleware in the industry. It offers the concept of functions, locks and semaphores. Hazelcast provides a distributed lock implementation and makes it possible to create a critical section within a cluster of JVM; so only a single thread from one of the JVM's in the cluster is allowed to acquire that lock. (Veentjer, 2013). Besides an Application Programming Interface (API) for asynchronous remote method invocations, Hazelcast has a simple API to store objects in a computer grid. JCL separates business logic from distribution issues and, in Hazelcast, both requirements are put together, so flexibility and dynamism are reduced during execution time. Hazelcast cannot instantiate a global variable remotely like JCL, i.e., it always maintains double copies of each variable at each remote instantiation. Hazelcast has manual scheduling for global variables and executions, so the end-user can control the cluster machine to store or run an algorithm. *Hazelcast* does not implement automatic deployment, so it is necessary to manually add each end-user class to the JVM classpath before starting each *Hazelcast* node.

Oracle Coherence is an in-memory data grid commercial middleware that offers database caching, HTTP session management, grid agent invocation and distributed queries (Seovic et al., 2010). Coherence provides an API for all services, including cache services and others. It enables an agent deployment mechanism, so there is the dynamic class loading feature in cluster nodes, but such agents must implement the EntryProcessor interface, thus refactorings are necessary.

RAFDA (Walker et al., 2003) is a reflective middleware. It permits arbitrary objects in an application to be dynamically exposed for remote access, allowing applications to be written without concern for distribution (Walker et al., 2003). RAFDA objects are exposed as Web services without requiring reengineering to provide distributed access to ordinary Java classes. Applications access RAFDA functionalities by calling methods on infrastructure objects named RAFDA runtime (RRT). Each RRT provides two interfaces to application programmers: one for local RRT accesses and the other for remote RRT accesses. RRT has peer-to-peer communication, so it is possible to execute a task in a specific cluster node, but if the end-user needs to submit several tasks to more than one remote RRT, a scheduler must be implemented from the scratch. RAFDA has no portable multi-core and multi-computer versions.

In the beginning of 2000's, an interesting middleware, named *FlexRMI*, was proposed by Taveira et al. (2003) to enable asynchronous remote method invocation using the standard Java Remote Method

Table 1: JCL and its counterparts' features.

Feature / Tool	Fault Tolerant	Refactoring required	Simple Deploy	Collaborative	Portable Code	Support Available
JCL	No	No	Yes	Yes	Yes	Yes
Infinispan	Yes	Low	No	Yes	No	Yes
JPPF	Yes	No	Yes	No	No	Yes
Hazelcast	Yes	Low	No	Yes	No	Yes
Oracle Coherence	Yes	Medium	*NF*[1]	Yes	No	Yes
RAFDA	No	No	Yes	Yes	No	Yes
PJ	No	*NF*[1]	*NF*[1]	No	Yes	Yes
FlexRMI	No	Medium	No	No	No	No
RMI	No	Medium	No	No	No	Yes
Gridgain	Yes	Low	No	Yes	No	Yes
ICE	Yes	High	No	No	No	Yes
MPJ Express	No	Medium	No	No	Yes	Yes
Jessica	*NF*[1]	No	Yes	No	Yes	Yes

1 - NF: Not found

Invocation (RMI) API. FlexRMI is a hybrid model allowing both asynchronous or synchronous remote methods invocations. There are no restrictions in the ways a method is invoked in a program. The same method can be called asynchronously at one point and synchronously at another point in the same application. It is the programmer's responsibility the decision on how the method call is to be made. (Taveira et al., 2003) FlexRMI changes Java RMI stub and skeleton compilers to achieve high transparency. *FlexRMI* is the RMI asynchronous, so there is no multi-core version. Furthermore, it requires at least "java.rmi.Remote" and "java.rmi.server.UnicastRemoteObject" extensions to produce a RMI application. Since it does not implement the dynamic class loading feature, all classes and interfaces must be stored in nodes before a RMI (and also FlexRMI) application starts, making deployment a time-consuming effort.

Gelibert et al. (2011) proposed a new middleware using Distributed Shared Memory (DSM) principles to efficiently simplify the clustering of dynamic services. The proposed approach consists in transparently integrating DSM into Open Services Gateway Initiative (OSGi) (OSGi, 2010) service model using containers and annotations. The authors use *Terracotta framework* (Terracotta Inc., 2008) as a kernel of the entire solution. Gelibert et al. (2011) point out limitations of using static types in the code, since instrumentation is done at runtime, thus the compiler cannot perform static verification on the application code. This creates complicated debugging scenarios when problems, especially transient ones, occur.

Programming Graphical Processing Unit (GPU) clusters with a distributed shared memory abstraction offered by a middleware layer is a promising solution for some specific problems, i.e. Single Instruction Multiple Data (SIMD) solutions. In (Karantasis and Polychronopoulos, 2011), an extension of *Pleiad* middleware (Karantasis and Polychronopoulos, 2009) is implemented enabling Java developers to work with a local GPU abstraction over several machines with one-four GPU devices each.

3 JCL ARCHITECTURE

This section details the architecture of the proposed middleware. The reflective capability of several programming languages, including Java, is an elegant way for middleware systems to introduce low coupling between distribution and business logic, as well as simplify the deployment process and introduce cloud multi-developer environments. Thus, reflection is the basis for many JCL features.

JCL has two versions: multi-computer and multi-core. The multi-computer version, named "Pacu", stores objects and executes tasks over a cluster where all communications are done via TCP/UDP protocol. On the other hand, the multi-core version, named

"Lambari", turns the User component into a local host component without the overhead of TCP/UDP communications. All objects and tasks are respectively stored and executed locally on the end-user machine. The architecture of JCL is composed of three main components: *User*, *Server*, and *Host*. While *User* is designed to expose the middleware services in a unique API to be adopted by developer, *Server* is responsible to manage the JCL cluster. Finally, the *Host* component is where the objects are stored and the registered methods are invoked. The next sections describe the design choices of JCL to provide the previously cited features.

3.1 User Component

To achieve portability, a single access point to JCL cluster is mandatory and the User component is responsible for that. It represents a unified API for both versions (multi-computer and multi-core) and it is where asynchronous remote method invocations and object storage take place. The user selects which version to start with according to a property file configured by the end-user. In the multi-core version, User avoids network protocols, performing shared memory communications with Host. In the multi-computer version, UDP and TCP/IP protocols are adopted, thus marshalling/unmarshalling, location, naming and several other components are introduced. These components are fundamental to distributed systems and are explained in details in Coulouris et al. (2007).

During the application execution, User component follows a pipeline composed of six steps: 1) receive end-user application calls; 2) generate unique identifiers for these calls; 3) return the identifiers to the end-user application; 4) schedule them; 5) submit them to Hosts; 6) and finally, store results of submitted calls locally. Since JCL is by default asynchronous, end-user application calls receive a ticket for each submitted task. After a complete execution of the pipeline above, the result is ready to be obtained using the identifier provided by User. Step 5 can be optimized in the multi-computer version if successive submissions occur, i.e. successive calls are buffered and submitted in batch to a Host.

This component adopts different strategies to schedule processing and storage calls in the multi-computer version. It allocates Hosts to handle processes according to the number of cores in the entire cluster. For instance, in a cluster with ten quad-core machines, we have forty cores, so User submits chunks of processing calls to the first machine, where each chunk size must be multiple of four, since it is a quadcore processor. Internally, a Host allocates a pool of threads, also with size multiple of four, to consume such processing calls. After the first chunk, User sends the second, the third and so on. After ten submissions, User starts submitting to the first machine again. The circular list behavior continues as long as there are processing calls to be executed. Heterogeneous clusters are possible, since JCL automatically allocates a number of chunks proportional to the number of cores of each machine.

There is a scheduling strategy for storage calls, so the User component calculates a function F to determine in which host the global variable will be stored (Equation 1), where $hash(vn)$ is the global variable name hashcode, nh is the number of JCL hosts and F is the node position. Experiments with incremental global variable names like "p_ij" or "p_i", where i and j are incremented for each variable and p is any prefix, showed that F achieves an almost uniform distribution for object storage over a cluster in several scenarios with different variable name combinations, however there is no guarantee of a uniform distribution for all scenarios. For this reason, User introduces a delta (d) property that normally ranges from $0-10\%$ of nh. The delta property relaxes function F result enabling two or more Hosts as alternatives to store a global variable.

$$F = \frac{|hash(vn)|}{nh} \tag{1}$$

In general, d relaxes a fixed JCL Host selection without introducing overhead in F. A drawback introduced by d is that JCL must check $(2*d)+1$ machines to search for an object, i.e., if d is equal to 2, JCL must check five machines (two machines before and two after the machine identified by function F in the logical ring). JCL checks all five alternatives in parallel, so the drawback is very small, as our experiments demonstrate. JCL with delta equals 2, 1 and 0 has similar execution time in clusters with 5, 10 and 15 machines.

3.2 Server Component

This component was designed to manage the cluster and is responsible for receiving the information from each Host and distributing it to all registered User components, enabling them to directly communicate with each Host. The Server also implements the possibility for the end-user to assign the placement of objects stored in the cluster, thereby disrespecting the Host selection obtained from the function F presented in Equation 1.

The Server fulfills the function of centralizing

component, receiving the features of the computer where each Host is installed. Before adding a new Host, the Server notifies its presence to all registered Hosts that, after receiving the new member registration notification, recalculate the function F and change the Host objects, if necessary. After all changes have taken place, the Server is notified, fulfilling the registration of the new Host.

When there is an end-user application running, at least one Host registration needs to be completed successfully, so that such application can receive the cluster map, enabling direct communication with the registered Hosts and eliminating the necessity to search for a Host in the Server at every new demand.

One of JCL's advantages is the possibility of storing Java objects in a specific Host. In this case, the end-user can specify Hosts that are different from those calculated by function F. To guarantee that all running applications in a specific cluster have access to all the instantiated objects, the locations assigned manually by the end-user are centralized in the Server, this way they do not depend on the function F. It is possible to note that the increase of manually assigned variables concentrates the workload on the Server, thus variables with high amount of accesses can cause bottlenecks in the Server component. The end-user can also choose the Host to execute their methods, therefore JCL scales or allows its developers to scale their demands.

3.3 Host Component

JCL Host has two basic functions: to store the objects sent by User or by another Host and invoke previously registered class methods. Its architecture allows the Host component to dynamically determine the number of threads (workers) running the end-user class methods. By default, the application is configured to use the total number of cores available on the Host, that is, if the computer has four cores, four workers are created. Nevertheless, the user can create as many workers as necessary by simply setting the property that assigns the number of threads to be created. The justification for such feature is the possibility to combine CPU-bound methods with I/O bound method executions, requiring the operating system to schedule them, what increases the number of context switches, however such extra workload usually pays off, since there is a possibility of prefetching CPU bound method executions while waiting for I/O results.

Before the Host component publishes its services to User components, it starts a JCL pipeline composed of three steps: 1) the Host notifies the Server its intention to join the cluster; 2) the Server propagates the existence of a new Host; 3) the Server allows the Host to join JCL cluster.

A property that differentiates JCL from most of its counterparts is the class registration process that simplifies deployment. This process, invoked by User and performed by Host, adds the necessary classes to JVM classpath at runtime, which enables the end-user to store objects and remotely execute methods without manually registering the class in each JVM of each cluster Host.

To perform object storage, two different alternatives are adopted. In the first one, the end-user creates the object and sends it to be stored in a Host, which may or may not be chosen by him. In the second one, the end-user defines which object should be created, passing its arguments for the constructor and, therefore, enabling the object instantiation directly at the Host. It is possible to note that the second storage option allows the creation of massive objects at Host without transferring them through the data network.

As described previously in this section, the middleware is based on Java reflection. This way, there is no need to adapt any classes so that they can be executed at Hosts. Once registered, the target classes, as well as their dependencies, are sent to Host where methods are mapped and available for execution. This feature enables JCL applications to separate business logic code from distribution code, as well as simplifies deployment and enables distributed objects storage.

4 USE CASE

This section aims to evaluate JCL in terms of fundamental computer science algorithms development, such as sorting. The JCL BIG sorting application was implemented, since it represents a solution with intensive communication, processing and I/O.

The distributed BIG sorting application is a sorting solution where data are partitioned and also sorted, i.e. there is no centralized sorting mechanism. Data are generated and stored in a binary file by each Host thread, performing parallel I/O on each Host component. Data are integers between -10^9 to $+10^9$. The final sorting contains one million different numbers and their frequencies distributed over a cluster, but the original input data were generated from two billion possibilities.

The sorting application is a simple and elegant sorting solution based on items frequencies. The frequency of each number of each input data partition is obtained locally by each Host thread and a chunk

```
24 JCL_facade jcl = JCL_FacadeImpl.getInstance();
25
26 int numJCLclusterCores = jcl.getClusterCores();
27 //registering
28 jcl.register(Random_Number.class, "Random_Number");
29
30 //builds the input data, partitioned over JCL cluster
31 Object[][] args = new Object[numJCLclusterCores][];
32 for(int i=0;i<numJCLclusterCores;i++) {
33    Object[] oneArg = {sementes, "output"+i};
34    args[i]= oneArg;
35 }
36 List<String> tickets = jcl.executeAllCores("Random_Number", "Create1GB",
       args);
37 jcl.getAllResultBlocking(tickets);
38 for(String aTicket:tickets) jcl.removeResult(aTicket);
39 tickets.clear();
40 tickets=null;
41 System.err.println("Time to create input (sec): " +
       (System.nanoTime()-time)/1000000000);
```

Figure 1: Main class - how to generate pseudo-random numbers in the JCL cluster.

strategy builds a local data partition for the entire JCL cluster, i.e. each thread knows how many JCL threads are alive, so all number frequencies (nf) divided by number of cluster threads (nct) create a constant C, i.e. $C = nf/nct$. Each different number in an input data partition is retrieved and its frequency is aggregated in a global frequency GF. When GF reaches C value, a new chunk is created, so C is fundamental to produce chunks with similar number frequencies without storing the same number multiple times. When JCL avoids equal number values it also reduces communication costs, since numbers of one Host thread must be sent to other threads in the JCL cluster to perform a fair distributed sorting solution.

The sorting is composed of three phases, besides the data generation and a validation phase to guarantee that all numbers from all input data partitions are retrieved and checked against JCL sorting distributed structure. The sorting has approximately 350 lines of code, three classes and only the main class must be a JCL class, i.e. inherit JCL behavior. The pseudo-aleatory number generation phase illustrates how JCL executes existing sequential Java classes on each Host thread with few instructions (Figure 1). Lines 24, 26 and 28 of the main class illustrate how to instantiate JCL, obtain JCL cluster number of cores and register a class named "Random_Number" in JCL, respectively. Lines 31-35 represent all arguments of all "Create1GB" method calls, so in our example we have "numJCLClusterCores" method arguments and each of them is a string labeled "output_suffix", where the suffix varies from 0 to "numJCLClusterCores" variable value.

Line 36 represents a list of tickets, adopted to store all JCL identifiers for all method calls, since JCL is by default asynchronous. The JCL method "executeAllCores" executes the same method "Create1GB" in all Host threads with unique arguments on each method call. Line 37 is a synchronization barrier, where big sorting main class waits until some tasks, identified by "tickets" variable, have finished. From lines 38-40 objects are destroyed locally and remotely (line 38), and finally in line 41 there is the time elapsed to generate pseudo-random numbers over a cluster of multi-core machines and in parallel. The "Random_Number" class is a sequential Java class and method "Create1GB" adopts Java Random math class to generate 1GB numbers on each input data partition binary file.

Phases one, two and three are similar to Figure 1, i.e. they are inside the main class and they behave basically splitting method calls over the cluster threads and then waiting all computations to end. Precisely, at phase one JCL reads the input and produces the set of chunks, as well as each chunk frequency or the frequencies of its numbers. C is calculated locally in phase one, i.e. for a single input data partition, so in C equation nf represents how many numbers an input data partition contains and nct represents the number of JCL Host threads. Phase one finishes its execution after storing all number frequencies locally in a JCL Host to avoid a second file scan. It is possible to note that phase one does not split the numbers across the local chunks, since the algorithm must ensure a global chunk decision for that.

After phase one, the main class constructs a global

```
169 long load=0; int b; String result = "";
170 for(Integer ac:sorted){
171     load+=map.get(ac);
172     if(load>(totalF/(numOfJCLThreads))){
173         b=ac;
174         result+=b+ ":";
175         load=0;
176     }
177 }
```

Figure 2: Main class - how to mount the global chunk schema to partition the cluster workload.

```
97 for(int r=0;r<numJCLThreads;r++) {
98 JCLMap<Integer, Map<Integer, Long>> h = new JCLHashMap<Integer, ↩
       Map<Integer,Long>>(String.valueOf(r));
99 h.put(id, final[r]);
```

Figure 3: Sorting class - how to deliver chunks to other Host threads.

sorting schema with fair workload. Figure 2 illustrates how the main class produces chunks with similar number frequencies. Each result of phase one contains a schema to partition the cluster workload, so a global schema decision must consider all numbers inside all chunks of phase one.

The main class calculates the total frequency of the entire cluster, since each thread in phase one also returns the chunk frequency. Variable "totalF" represents such a value. Lines 169 to 177 represent how JCL sorting produces similar chunks with a constant C as a threshold. The global schema is submitted to JCL Host threads and phase two starts.

Phase two starts JCL Host threads and each thread can obtain the map of numbers and their frequencies, generated and stored at phase one. The algorithm just scans all numbers and inserts them into specific chunks according to the global schema received previously. Phase two ends after inserting all numbers and their frequencies into JCL cluster to enable any JCL Host thread to access them transparently. Figure 3 illustrates JCL global variable concept, where Java objects lifecycles are transparently managed by JCL over a cluster. The sorting class obtains a global JCL map labelled "h" (Figure 3). Each JCL map ranges from 0 to number of JCL threads in the cluster (line 97), so each thread manages a map with its numbers and frequencies, where each map entry is a chunk of other JCL Host thread, i.e. each JCL Host thread has several chunks created from the remaining threads. Line 99 of Figure 3 represents a single entry in a global map "h", where "id" represents the current JCL Host thread identification and "final" variable represents the numbers/frequencies of such a chunk. Phase three of sorting application just merges all chunks into a unique chunk per JCL Host thread. This way, JCL guarantees that all numbers are sorted, but not centralized in a Server or Host component, for instance.

Our sorting experiments were conducted with JCL multi-computer version. The first set of experiments evaluated JCL in a desktop cluster composed of 15 machines, where 5 machines were equipped with Intel I7-3770 3.4GHz processors (4 physical cores and 8 cores with hyper-threading technology) and 16GB of of RAM DDR 1333Mhz, and the other 10 machines were equipped with Intel I3-2120 3.3GHz processors (2 physical cores and 4 cores with hyper-threading technology) and 8GB of RAM DDR 1333Mhz. The Operating System was a Ubuntu 14.04.1 LTS 64 bits kernel 3.13.0-39-generic and all experiments could fit in RAM memory. Each experiment was repeated five times and both higher and lower runtimes were removed. An average time was calculated from the three remaining runtime values. JCL distributed BIG Sorting Application version sorted 1 TB in 2015 seconds and the OpenMPI version took 2121 seconds, being JCL 106 seconds faster. Both distributed BIG sorting applications (JCL and MPI) implement the explained idea and are available at JCL website.

The second experiments evaluated JCL in an embedded cluster composed of two raspberry pi devices, each one with an Arm ARM1176JZF-S processor, 512MB of RAM and 8GB of external memory, and one raspberry pi 2 with a quadcore processor operating at 900MHz, 1GB of RAM and 8GB of external memory. The Operating System was Raspbian Wheezy and all experiments could fit in RAM memory. Each experiment was repeated five times and both higher and lower runtimes were removed. An

average time was calculated from the three remaining runtime values.

The JCL distributed BIG sorting was modified to enable devices with low disk capacity to also sort a big amount of data. Basically, the new sorting version does not store the pseudo-random numbers in external memory. It gathers the number generation phase with the phase where number frequencies are calculated. Differently from other IoT middleware systems (Perera et al., 2014), where small devices such as raspberry pi are adopted only for sensing, JCL introduces the possibility to implement general purpose applications and not only sensing ones. Furthermore, JCL sorting can run on large or small clusters, as well as massive muti-core machines with a unique portable code. The small raspberry pi cluster sorted 60GB of data in 2,7 hours.

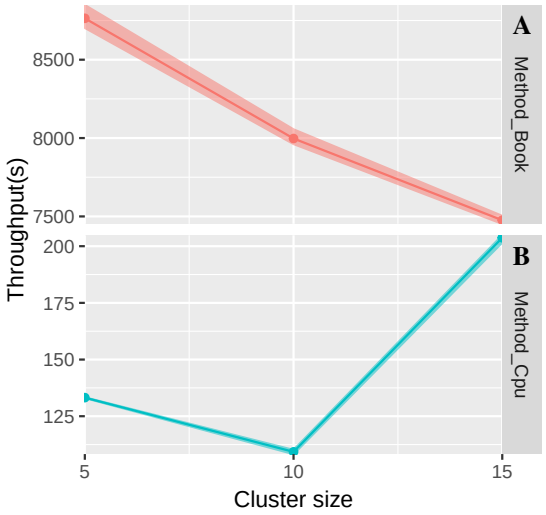


Figure 4: Method invocation.

5 EXPERIMENTAL EVALUATION

Experiments were conducted with JCL multi-computer and multi-core versions. Initially, the JCL middleware was evaluated in a desktop cluster composed of 15 machines, the same cluster used to test JCL distributed BIG sorting application. The middleware was evaluated in terms of throughput, i.e., the number of processed JCL operations per second. The goal of these experiments is to stress JCL measuring how many executions it supports per second and also how uniform function F, presented in Equation 1, can be when both incremented global variable names and random names are adopted.

In the first set of experiments, we tested JCL asynchronous remote method invocation (Figure 4). For each test we fixed the number of remote method invocations to 100 thousand executions. JCL Protocol Buffer Algorithm (PBA) algorithm was adopted, so JCL differentiates the sizes of both machines of the cluster to configure the workload. The experiments were composed of two different methods: the first one is a void method with a book as argument, where a book is a user type class composed of authors, editors, edition, pages and year attributes (Figure 4 A); and the second method is composed of an array of string and two integer values as arguments which are adopted by algorithms for calculating Levenshtein distance, Fibonacci series and prime numbers (Figure 4 B). We measured the throughput of each cluster configuration (5, 10 and 15 machines).

The results demonstrated that JCL's throughput rises when cluster size increases as the task becomes more CPU bound. There is a throughput decrease when the cluster increases from 5 to 10 machines and

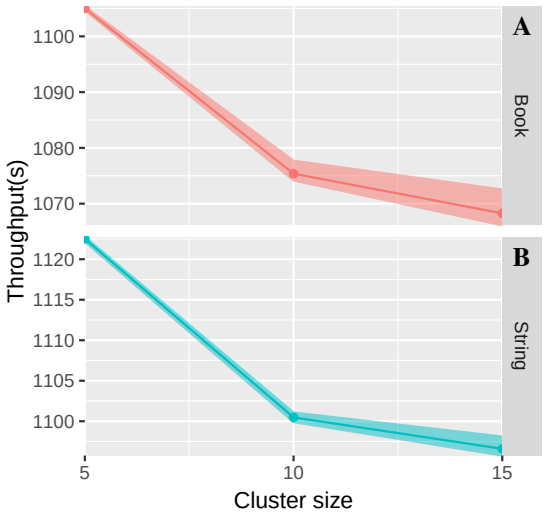


Figure 5: Global variable experiments.

the justification is that the new five Hosts are smaller than the first five ones, so the throughput does not increase at the same rate (Figure 4 B). The second test represents non CPU bound scenarios, so it is clear that network overhead is greater than task processing (Figure 4 A).

In the second set of experiments (Figure 5), we fixed the number of instantiated global variables to 40 thousand instantiations. We tested the book class instantiation explained previously (Figure 5 A) and also a smaller object like a string with 10 characters (Figure 5 B). We tested the five best machines first and then added the ten worst machines, thus this strategy may influence the throughput results negatively. As the cluster enlarges, the number of connections and other issues also become time-consuming, thus a re-

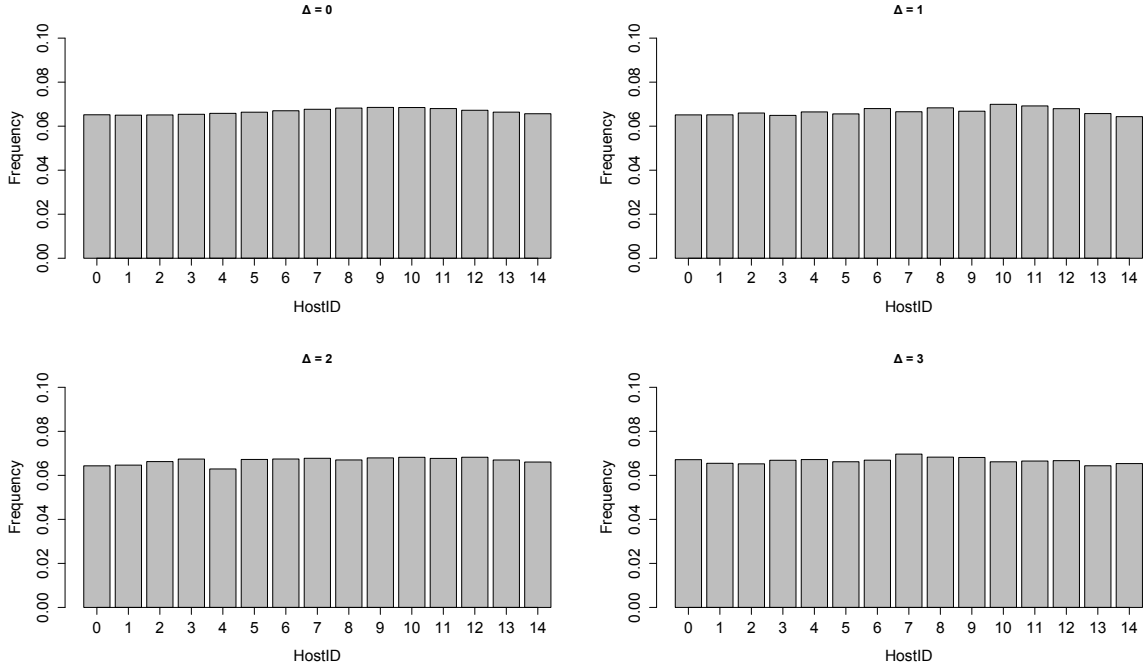


Figure 6: Variable names with autoincrement.

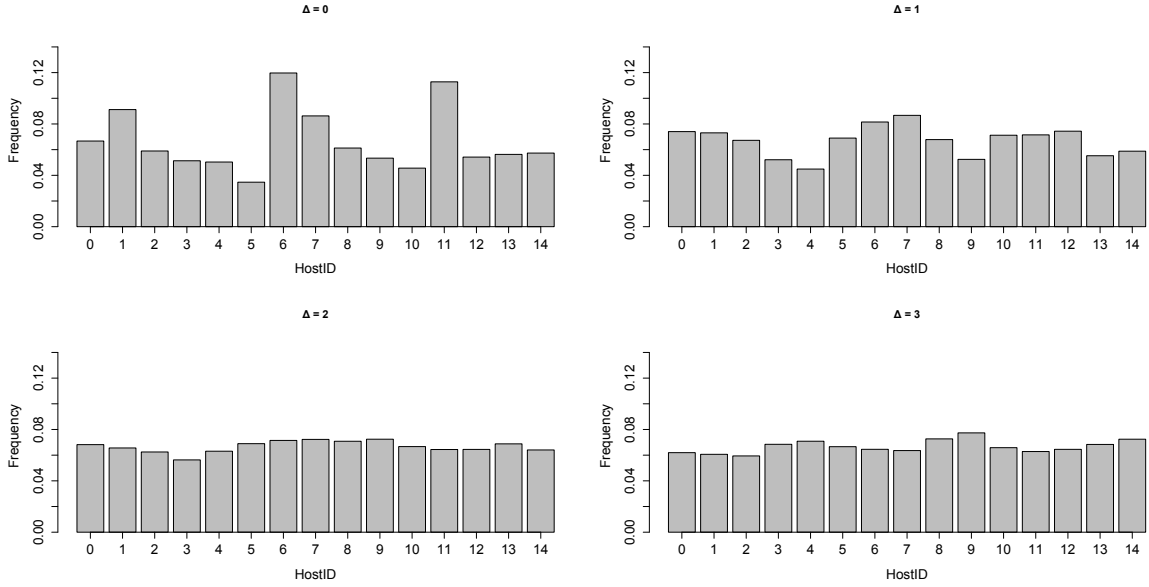


Figure 7: Bag of words.

duction in throughput should be expected. Another important observation is the synchronous behavior of JCL shared memory services, which is another bottleneck when the cluster becomes bigger. The results demonstrate that as the global variable becomes more complex, the data network overhead increases. Precisely, the throughput of String variable reduces 2,3% when JCL cluster increases from 5 to 15 Hosts, but when book variable is adopted, the throughput reduces 3,3% in the same cluster configurations. The positive aspect of such a scenario is that JCL over bigger clusters stores more data.

In the third set of experiments, we evaluated the uniformity of function F presented in Equation 1 plus a delta d and instantiated 40 thousand variable names. JCL cluster size was set to 15 machines, and then we tested different prefix variable names, and also autoincrement suffixes, i.e., variable names like "p_i" and

"*p_i j*", where p is a prefix and i and j are autoincrement values. We also tested $F + d$ distribution for an arbitrary bag of words and chose the Holy Bible's words to verify how JCL data partition performs. The results are illustrated in Figure 6 and 7, where Δ is delta size. Usually, JCL achieves an almost uniform distribution using delta between zero and two.

The result of the bag of arbitrary words becomes more uniform as delta increases, so even when the end-user decides to adopt arbitrary variable names in the code, JCL can achieve an almost fair data partition. We also tested the JCL overhead when a variable content is retrieved using delta zero, one and two, and there are almost no overhead varying deltas, but data partition uniformity is reduced as delta tends to zero, what can be seen in Figure 7. The justification is that network communication times for data checkings are irrelevant when compared with remote instantiation runtimes.

Finally, we also evaluated JCL multi-core version against a Java thread implementation provided by Oracle. An Intel I7-3770 3.4GHz processor with 8 cores, including hyper-threading technology, and 16GB of RAM was used in the experiment. We implemented a sequential version for a CPU bound task composed of existing Java sequential algorithms for calculating Levenshtein distance, Fibonacci series and prime numbers. We calculated JCL and Java threads speedups, and the results demonstrated similar speedups, i.e. in a machine with four physical cores and four virtual cores, JCL achieved speedup of 5.61 and Oracle Java threads the speedup of 5.77.

6 CONCLUSION

In this paper, we present a novel reflective middleware that is able to invoke remote methods asynchronously and also manage Java objects lifecycle over a cluster of JVMs. JCL is designed for multi-core, multi-computer and hybrid computer architectures. End-users write portable JCL applications, where global variables are also multi-developer, so different applications can transparently share resources without explicit references over a computer cluster. JCL can execute existing Java code or JCL code, this way JCL can build complex applications. Reflection capabilities enable JCL to separate distribution from business logic, enabling both existing sequential code executions over many high performance computer architectures with zero changes and multiple distribution strategies for a single sequential algorithm according to a hardware specification. Deployment in JCL is not time consuming, i.e. a JCL cluster without end-user code is sufficient to run any Java application in JCL. No other middleware solution puts all these features together in a unique solution.

Experiments demonstrate that JCL is a promising solution, although many improvements must be done. JCL must implement security methods. End-users should be able to lock/unlock global variables and execute tasks from private groups over a single JCL cluster, enabling different collaboration levels or profiles. JCL must be fault tolerant in storage and processing. Future systems should be able to recover on their own. Self-stabilization, self-healing, self-reconfiguration and recovery-oriented computing implement several algorithms/protocols that can be incorporated into JCL. JCL must implement the concept of multi-server, therefore a JCL server can manage, for instance, a cluster of JCL hosts with invalid IPs and communicate with other JCL servers, providing a multi-cluster JCL solution. GPU execution abstractions, where location and copies are transparent to end-users, are fundamental to JCL. A heuristic based scheduler, where cloud requirements are considered, is also an important improvement to JCL. An API for sensing is fundamental to JCL for IoT. Cross-platform Host component, including platforms without JVM, with JVMs that are not compatible with JSR 901 (Java Language Specification) or platforms without operating system, are mandatory to IoT. Built-in modules for monitoring and administration should be added to JCL.

ACKNOWLEDGEMENTS

We thank José Estevão Eugênio de Resende and Gustavo Silva Paiva for helping with experiments, Universidade Federal de Ouro Preto (UFOP) and Instituto Federal de Minas Gerais (IFMG) for the infrastructure, and Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG) for financial support.

REFERENCES

Brynjolfsson, E. M. (2012). Big data: The management revolution. *Harvard Business Review*, 90(10):6066.

Cohen, L. (2015). *Java Parallel Processing Framework*. Available from: ⟨http://www.jppf.org/⟩.[15 Dezember 2015].

Coulouris, G., Dollimore, J., and Kindberg, T. (2007). *Sistemas Distribuídos - 4ed: Conceitos e Projeto*. Bookman Companhia.

Di Sanzo, P., Quaglia, F., Ciciani, B., Pellegrini, A., Didona, D., Romano, P., Palmieri, R., and Peluso, S.

(2014). A flexible framework for accurate simulation of cloud in-memory data stores. *arXiv preprint arXiv:1411.7910*.

Egan, S. (2005). *Open Source Messaging Application Development: Building and Extending Gaim*. Apress.

Forum, M. P. (1994). Mpi: A message-passing interface standard. Technical report, Knoxville, TN, USA.

Gelibert, A., Rudametkin, W., Donsez, D., and Jean, S. (2011). Clustering osgi applications using distributed shared memory. In *Proceedings of International Conference on New Technologies of Distributed Systems (NOTERE 2011)*, pages 1–8.

Ghosh, S. (2014). *Distributed systems: an algorithmic approach*. CRC press.

Gokhale, A., Balasubramanian, K., Krishna, A. S., Balasubramanian, J., Edwards, G., Deng, G., Turkay, E., Parsons, J., and Schmidt, D. C. (2008). Model driven middleware: A new paradigm for developing distributed real-time and embedded systems. *Science of Computer programming*, 73(1):39–58.

Han, J., Kamber, M., and Pei, J. (2011). *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition.

Henning, M. and Spruiell, M. (2006). Distributed programming with ice reading.

Hindman, B., Konwinski, A., Zaharia, M., Ghodsi, A., Joseph, A. D., Katz, R., Shenker, S., and Stoica, I. (2011). Mesos: A platform for fine-grained resource sharing in the data center. In *Proceedings of the USENIX Conference on Networked Systems Design and Implementation (NSDI 2011)*, pages 295–308.

Kaminsky, A. (2015). *Big CPU, Big Data: Solving the World's Toughest Computational Problems with Parallel Computing*. Unpublished manuscript. Retrieved from http://www.cs.rit.edu/˜ark/bcbd.

Karantasis, K. and Polychronopoulos, E. (2011). Programming gpu clusters with shared memory abstraction in software. In *Proceedings of Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP 2011)*, pages 223–230.

Karantasis, K. I. and Polychronopoulos, E. D. (2009). Pleiad: A cross-environment middleware providing efficient multithreading on clusters. In *Proceedings of ACM Conference on Computing Frontiers (CF 2009)*, pages 109–116.

Murphy, A. L., Picco, G. P., and Roman, G.-C. (2006). Lime: A coordination model and middleware supporting mobility of hosts and agents. *ACM Trans. Softw. Eng. Methodol.*, 15(3):279–328.

Nester, C., Philippsen, M., and Haumacher, B. (1999). A more efficient rmi for java. In *Proceedings of the ACM 1999 conference on Java Grande*, pages 152–159. ACM.

OSGi (2010). Osgi specification release 4.2.

Perera, C., Liu, C. H., Jayawardena, S., and Chen, M. (2014). A survey on internet of things from industrial market perspective. *Access, IEEE*, 2:1660–1679.

Pitt, E. and McNiff, K. (2001). *Java.Rmi: The Remote Method Invocation Guide*. Addison-Wesley Longman Publishing Co., Inc.

Seovic, A., Falco, M., and Peralta, P. (2010). *Oracle Coherence 3.5*. Packt Publishing Ltd.

Taboada, G. L., Ramos, S., Expósito, R. R., Touriño, J., and Doallo, R. (2013). Java in the high performance computing arena: Research, practice and experience. *Science of Computer Programming*, 78(5):425–444.

Taboada, G. L., Touriño, J., and Doallo, R. (2009). Java for high performance computing: assessment of current research and practice. In *Proceedings of the 7th International Conference on Principles and Practice of Programming in Java*, pages 30–39. ACM.

Tanenbaum, A. S. and Van Steen, M. (2007). *Distributed systems*. Prentice-Hall.

Tariq, M. A., Koldehofe, B., Bhowmik, S., and Rothermel, K. (2014). Pleroma: a sdn-based high performance publish/subscribe middleware. In *Proceedings of the 15th International Middleware Conference*, pages 217–228. ACM.

Taveira, W. F., de Oliveira Valente, M. T., da Silva Bigonha, M. A., and da Silva Bigonha, R. (2003). Asynchronous remote method invocation in java. *Journal of Universal Computer Science*, 9(8):761–775.

Team, I. (2015). *Infinispan 8.1 Documentation*. Available from: ⟨http://infinispan.org/docs/8.1.x/index.html⟩.[15 Dezember 2015].

Terracotta Inc. (2008). *The Definitive Guide to Terracotta: Cluster the JVM for Spring, Hibernate and POJO Scalability*. Springer Science & Business.

Troester, M. (2012). Big data meets big data analytics. *Cary, NC: SAS Institute Inc*.

Veentjer, P. (2013). *Mastering Hazelcast*. Hazelcast.

Walker, S. M., Dearle, A., Norcross, S. J., Kirby, G. N. C., and McCarthy, A. (2003). Rafda: A policy-aware middleware supporting the flexible separation of application logic from distribution. Technical report, University of St Andrews. Technical Report CS/06/2.

Watson, R. T., Wynn, D., and Boudreau, M.-C. (2005). Jboss: The evolution of professional open source software. *MIS Quarterly Executive*, 4(3):329–341.

Xiong, J., Wang, J., and Xu, J. (2010). Research of distributed parallel information retrieval based on jppf. In *2010 International Conference of Information Science and Management Engineering*, pages 109–111. IEEE.

Zaharia, M., Chowdhury, M., Franklin, M. J., Shenker, S., and Stoica, I. (2010). Spark: cluster computing with working sets. In *Proceedings of the 2nd USENIX conference on Hot topics in cloud computing*, pages 10–10.

Zhang, Q., Cheng, L., and Boutaba, R. (2010). Cloud computing: state-of-the-art and research challenges. *Journal of internet services and applications*, 1(1):7–18.

SHORT PAPERS

Evaluating the Teaching of Project Management Tools through a Series of Case Studies

Rafael Queiroz Gonçalves and Christiane Gresse von Wangenheim
Department of Informatics and Statistics, Graduate Program on Computer Science, Federal University of Santa Catarina (UFSC), Florianópolis, SC, Brazil
rafael.queiroz@posgrad.ufsc.br, c.wangenheim@ufsc.br

Keywords: Project Management, PMBOK, Instructional Design, Instructional Unit, Dotproject+.

Abstract: Project management (PM) tools are mandatory to properly manage software projects. The usage of these tools is an important competence for professionals in the computer area, and its teaching is addressed in superior computer courses. In this context, general usage tools are usually adopted, such as MS-Project, but the lack of educational features of in these tools has motivated the development of several educational PM tools. However, previous studies have shown that these tools still do not cover the whole PM process, as defined by PMBOK. As a result, this study aims at presenting an instructional unit to assist in the teaching of functionalities that support an extensive part of this process, covering the initiating and planning processes groups for all knowledge areas. It adopts an open-source and educational PM tool – dotProject+, and other instructional materials. The instructional unit was applied in several case studies in undergraduate computer courses. Its results demonstrated students were able to learn how to use the PM tool to carry out that part of PM process, and teachers state students learning was facilitated by the instructional materials adoption.

1 INTRODUCTION

Project Management (PM) is an important area for many organizations in the software industry. It is so because several projects still fail due to a lack of proper management, leading to problems related to unaccomplished deadlines, budget overrun, or scope coverage (The Standish Group, 2013). In this context, a project is defined as a temporary endeavor to achieve a single result, and PM is the use of knowledge, abilities, tools, and techniques that enable a project to reach its goals (PMI, 2013).

Project problems take place mainly because of the absence of a PM process (Keil et al., 2003), resulting in a narrow control over project restrictions (The Standish Group, 2013). The adoption of a PM process may be aided by the usage of a PM tool (Fabac et al., 2010). Despite many organizations still not using any PM tool, the positive contributions that these tools have brought about have increased the interest in their usage (Cicibas et al., 2010).

Given that the usage of PM tools is not well-established in organizations and that projects still fail, a possible cause for this may be the lack of teaching project managers and team members in the usage of these tools (The Standish Group, 2013; Fabac et al., 2010; Reid and Wilson, 2007).

The teaching of PM has to address the knowledge on PM, beyond general knowledge on administration, project environment, and interpersonal abilities (PMI, 2013). However, the teaching of PM should not just be focused on theoretical knowledge, because this is not enough to employ the PM effectively. It is crucial to develop the project manager competencies, which include knowledge (theoretical), abilities (practical), and attitudes (Branch., 2009). In addition to this, the PM is infeasible without the support of a PM tool, due to the complexity of contemporary software projects. Furthermore, the usage of these tools is among the project manager competencies (PMI, 2013; Salas-Morera et al., 2013). A PM tool is a software that supports the whole PM process or just a specific part of it. Among its supported functionalities are: schedule development, cost planning, risk analysis, etc. (Car et al., 2007).

However, there are a wide variety of PM tools, and most of them are not suitable for teaching, complicating the learning of their usage (PMI, 2013; Keil et al., 2003). For instance, some PM tools demand an initial effort to setup the environment and to learn about its usage, thus being rejected by some students

Gonçalves, R. and Wangenheim, C.
Evaluating the Teaching of Project Management Tools through a Series of Case Studies.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 393-404
ISBN: 978-989-758-187-8

during their first contact with them (Salas-Morera et al., 2013).

As an attempt to improve this scenario, some research (Reid and Wilson, 2007; Gregoriou et al., 2010) has identified that MS-Project is the most adopted PM tool for teaching. However, the lack of didactic features in this tool has motivated these research which proposes new educational PM tools (Salas-Morera et al., 2013). Most of these tools are focused on specific techniques, such as CPM, PERT, RACI Matrix. Nonetheless, when considering the PM process, none of these tools have focused on addressing of all PM knowledge areas. Thus, this paper presents an Instructional Unit (IU) that adopts the educational PM tool – dotProject+ – for teaching the usage of functionalities that supporting the execution of all PM knowledge areas for the initiating and planning processes groups.

An IU is a set of classes designed to teach certain learning objectives for a specific target audience. It consists of a set of instructional materials, for teachers and students, which are developed to enable the learning in a specific educational context (Hill et al., 2005).

The paper structure presents, in the background section, the main concepts related to software PM, PM tools, and teaching of PM tools. Section 3 presents related studies, and Section 4 presents the IU for teaching PM tools, including its instructional materials, such as the educational PM tool dotProject+. Section 5 presents the case study definition, which instances are presented in Section 6. In Section 7, we present the IU evaluation, along with a discussion of the research results, leading to the paper's conclusions in Section 8.

2 BACKGROUND

Concepts that are relevant to this research are presented in this section. All these concepts are utilized during the presentation of the IU and in the discussion of the case studies results.

2.1 Project Management

The PM directs the project activities and its resources in order to meet the project requirements. It is organized in 5 processes groups, which guide the PM process from its initiating to its closing (Figure 1).

Orthogonally to the processes groups, the PM processes are organized in 10 knowledge areas (Table 1), which may be addressed to effectively manage a project.

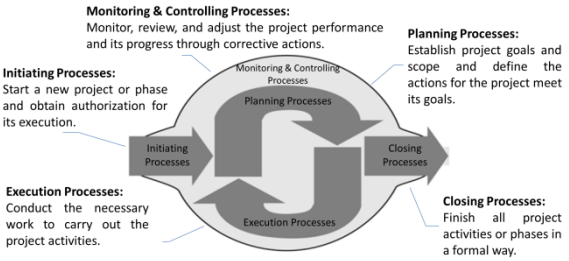


Figure 1: PM processes groups (PMI, 2013).

Table 1: PM knowledge areas (PMI, 2013).

Knowledge area	**Processes to:**
Integration	Identify and coordinate PM processes and PM activities.
Scope	Ensure that the project addresses the entire work to meet its requirements.
Time	Plan and control the activities that will be carried out during the project, so it concludes within the deadline.
Cost	Plan, estimate, and control project costs, so it concludes within the approved budget.
Quality	Define the goals, and quality policies, so the project meets the needs that have initiated it.
HR	Organize and manage the project team.
Communication	Ensure the generation, collection, and distribution of project information.
Risk	Identify and control the project risks.
Acquisition	Buy or contract products, services or any resources that are not available as project internal resources.
Stakeholder	Identify and manage the stakeholders and its expectations.

In the context of this study, the PM process refers to the one defined by PMBOK (PMI, 2013), which is the main reference in this area and is widely accepted worldwide (Ojeda and Reusch, 2013). The application of a PM process is aided by the usage of PM tools, which take advantage from technology either to support the whole PM process, or a specific part of it. This support may semi-automatize a few activities of the PM process, such as the schedule development, registering the project activities and its sequencing, and providing online forms to record their estimated durations and resources, then compiling its result in a gantt chart, instead of performing all the work manually (Cicibas et al., 2010). On the other hand, some PM process activities may be totally automated by PM tools, for instance, the calculi of project total cost, the critical path method identification, or its over allocated resources

(Fabac et al., 2010; Gregoriou et al., 2010).

2.2 PM Tools

Carrying out the PM process may be very complex, and demand many organizational resources. To assist its execution, many PM tools have been developed. Examples of PM tools are: MS-Project (microsoft.com/project), GanttProject (ganttproject.biz), DotProject (dotproject.net), Project.net (project.net), etc. (Fabac et al., 2010; Mishra, 2013).

However, due to the wide variety of PM tools, their functionalities and characteristics are very heterogenic (Pereira et al. 2013). The supported functionalities, for example, may cover the whole PM process, or just one or a few PM knowledge areas, or even more specifically just some activities, such as the tracking of worked hours or registering the project stakeholders.

Beyond its functionalities, other features may also influence the choice of the PM tool to be adopted for teaching. According to its features, some particularities of computational environment may be demanded, besides economic investments. Among these features, the most relevant are: availability, platform and usage propose.

The PM tools availability may be proprietary (the use of a license or acquisition is mandatory and it is maintained exclusively by a single organization) or open-source (free usage and maintained by users community). The proprietary PM tools may be adopted just by organizations that are prepared to perform its acquisition, while others may prefer to adopt an open-source tool, as a more economically savy alternative.

In terms of platform, there are the stand-alone tools (mono-user and desktop access) or web-based (multi-user and web browser access). In practice, a web-based PM tool has to be adopted to properly manage a software project, because it promotes collaborative work and information sharing (Cicibas et al., 2010). Thus, the teaching of these tools better prepares the student for a professional career (Reid and Wilson, 2007). However, the adoption of a PM web-based tool requires its installation in a web server that complies with the tool specification, and where internet access is provided to students.

Beyond the general usage PM tools, such as MS-Project or DotProject, that are focused on the professional daily routine, there are educational PM tools, which focus on student learning. These tools include didactic features, such as instructions about the usage of its functionalities, and simulations which create scenarios that facilitate the usage of specific PM techniques. Some examples of educational PM tools are DrProject, ProMES and PpcProject (Gregoriou et al., 2010).

2.3 Teaching of PM Tools

The usage of PM tools figures among the project manager competencies (PMI, 2013). The need for teaching this competency is addressed by the ACM/IEEE reference curriculum for Computer Science (ACM and IEEE, 2013). It specifies that students have to develop knowledge in all PM knowledge areas, and have to learn the usage of a PM tool to develop a project schedule, to perform risk analysis, to monitor the project performance, etc. Often the teaching of PM tools includes the application of the following techniques (PMI, 2013; Reid and Wilson, 2007; Gregoriou et al., 2010): the Critical Path Method (CPM) – that identifies the project activities that cannot be delayed without affecting the project deadline; the Program Evaluation and Review Technique (PERT) – that calculates the estimated effort to carry out an activity based on three other estimates (worst case, most common case, and best case); the RACI Matrix - describes the participation by various roles in completing project activities; the Resources Levelling - technique in which start and finish dates are adjusted based on resource constraints, with the goal of balancing demand for resources with the available supply; amongst others.

3 RELATED STUDIES

Related studies have been identified by previous research (Gonçalves and Wangenheim, 2015) which has presented some IUs that adopts educational PM tools. Among all the studies that have been found, we have selected just the ones which present the IU evaluation through some case study with students in undergraduate computer courses.

3.1 DrProject

This related study (Reid and Wilson, 2007) presents an IU that make use of an educational PM tool, DrProject, that is open-source and web-based. This PM tool proposes to be simple enough to be learned in just a few hours, but covering several functionalities of PM tools. The employed strategy is focused on collaborative project development, involving students groups through PM tools

functionalities. This IU was applied during 4 semesters in Software Engineering disciplines. It begins with an expositive class, providing theoretical instructions about time, human resources, and communication management, and its support through PM tool functionalities. Afterwards, the class is organized into groups composed of 4 to 5 students, that have to develop a software project. The groups work on the project during the semester, and use DrProject to develop the project schedule, organize and share the project artifacts, and also to carry out the whole communication among group members. This IU was evaluated to identify whether the students considered it was simple to learn about the usage of a PM tool with the support of DrProject. The data collection occurred at the end of each semester, by students answering a questionnaire. Its analysis demonstrated that 2/3 of students considered PM tool usage simple to learn. Some general comments highlighted that students complained about the lack of a tutorial explaining DrProject functionalities. Another part of the evaluation was based on teacher observation, that highlighted the PM tool has motivated the students to produce the project artifacts with more quality, and has facilitated the collaborative work among the students.

3.2 ProMES

This related study (Gregoriou et al., 2010) presents an IU that uses the educational PM tool, ProMES, which is open-source and stand-alone, for teaching CPM, PERT and RACI matrix techniques. The instructional strategy is based on scenarios (problems) resolution. In each scenario the student has to solve a problem using a specific technique, and when it is solved, another one is presented with a higher level of difficulty. This tool includes some educational features, such as the configuration of student level of experience, namely: trainee and professional. At the trainee level the PM tool presents feedback, assisting the student to identify each error, conducting him to the scenario resolution. On the other hand, the professional level does not provide any assistance. Another instructional feature of this tool is the tutorial video that is presented when the student first accesses the tool, explaining how to use its functionalities. The usage of this tool had been evaluated by teacher observation, and also by collecting verbal feedback from students. It was applied with 121 students during 3 semesters. It leads to conclusions that the ProMES promoted PM learning, highlighting the benefits of its educational features.

3.3 PpcProject

This related study (Salas-Morera et al., 2013) presents an IU that adopts the educational PM tool, PpcProject, which is open-source and stand-alone, and is focused on the teaching of CPM, PERT and resources allocation techniques. This tool has been developed to fulfil the same requirements provided by MS-Project when it is adopted for teaching, but to be superior for educational proposes. This IU was evaluated to verify whether students prefer to learn using PpcProject or MS-Project. It has been conducted through an experiment involving a total of 54 students. They were organized in two groups, control and experimental groups. Each group carried out the same activities, one using PpcProject, and other using MS-Project. In a second stage, each group carried out again the same activities, but using the other tool. Thus, each student has responded to 24 questions (12 for each PM tool). Their answers have been analysed by a non-parametric statistic test. This analysis has demonstrated that PpcProject is more suitable for teaching then MS-Project, except for the resources allocation process.

Analysing the IUs presented in the related studies, it is identified that the adopted educational PM tools have contributed for students learning. The assistance these tools provide have facilitated the content understanding, beyond facilitating the PM tool usage in class room. However, the IUs learning goals are generally focused on time and human resources management. Thus, considering the whole PM process, still there is a huge gap of what is currently been taught and all PM knowledge areas.

4 IU FOR TEACHING PM TOOLS TO SUPPORT THE PM PROCESS

In this context, this section presents an IU for teaching PM tools focused on initiating and planning processes groups, covering all knowledge areas. We have focused on these processes groups because they may be carried out within the IU discipline hours.

The execution of the planned projects may demand more hours than available, especially in the

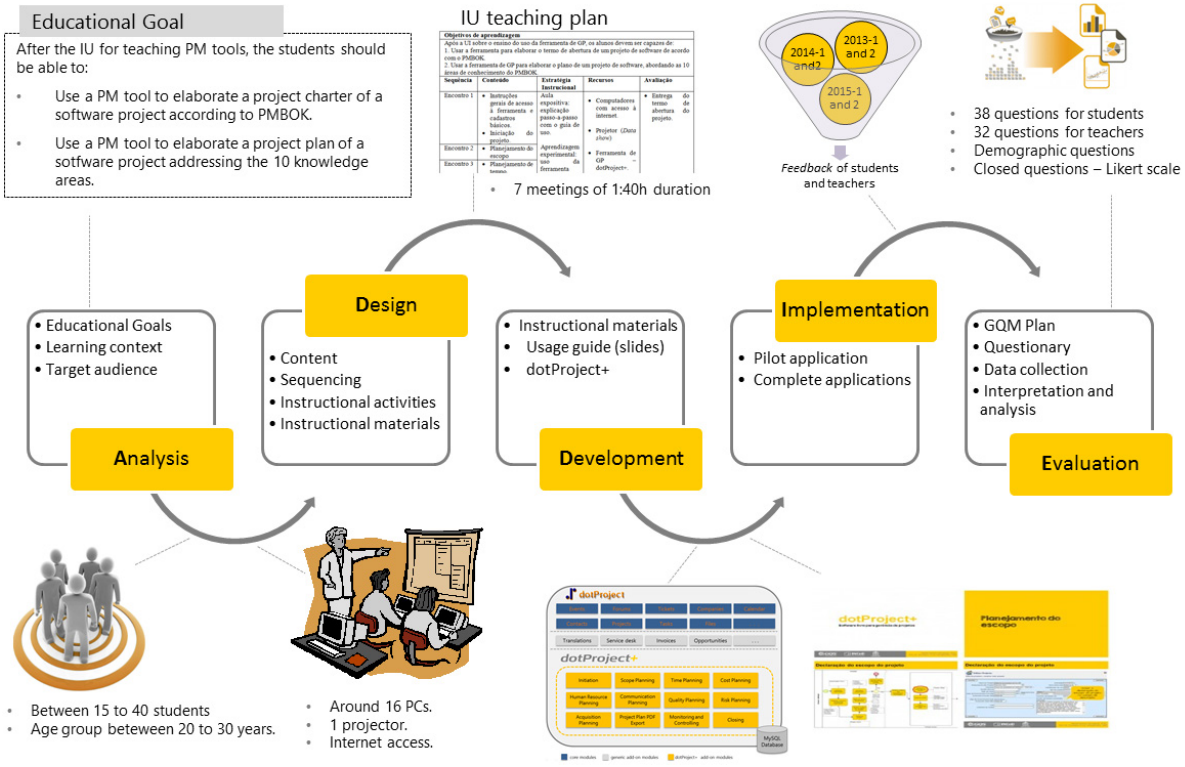


Figure 2: Execution of ADDIE process.

IU context, which projects are related to students term paper, demanding about a year to be concluded.

Effective and motivating IUs are developed following an Instructional Design process, for instance, ADDIE (Branch, 2009). An overview of the ADDIE process for the development of the proposed IU is presented in Figure 2.

As presented in Figure 2, the ADDIE process has 5 phases. Firstly, in analysis phase it is identified the target audience and the learning environment. This phase also includes the IU educational goals definition. Then, in analysis phase, it is defined the content to be addressed and its sequencing. This content is grouped in one or more meetings, and with the definition of instructional materials and activities, it composes the IU teaching plan. In the development phase, the instructional materials are developed, then leading to the implementation phase, which performs the IU application in class room. To evaluate the IU quality it is necessary to perform observations and data collection about teachers and students perception about the IU.

The next sections present details of instructional materials and about the IU evaluation process.

4.1 DotProject+

DotProject is one of most popular open-source tools for PM (Mishra, 2013). And previous studies have identified that among open-source alternatives, it is the most aligned with PMBOK (Pereira et al. 2013). DotProject architecture is organized in core modules, developed by its core team, and add-on modules, developed by users' community, which, may be installed on demand. Thus, we decided to adopt this tool, not only because its wide coverage of the PM process, but also because its functionalities may be extended via add-on modules. In this context, dotProject+ was developed, being composed by dotProject core modules, and several add-on modules that have been developed to enhance dotProject functionalities to cover all PM knowledge areas and also include didactic features (Figure 3). An example of these features is the related to the organization of its functionalities, which are hierarchically grouped by processes groups, and then by knowledge areas. Thus, when the student is using a certain functionality, it is easy to identify what part of PM process are been supported by each functionality.

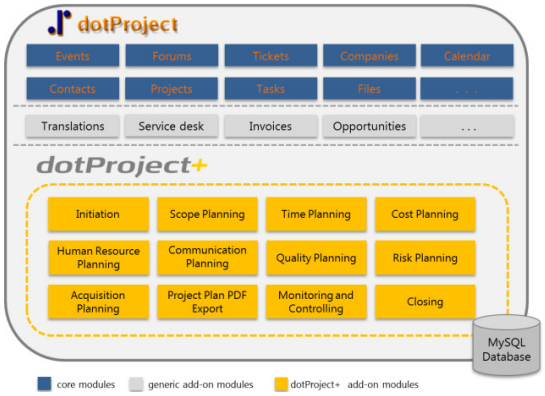


Figure 3: dotProject+ architecture.

Beyond adopting dotProject+, the IU also adopts the dotProject+ usage guide. This material is organized as presentation slides, which may serve to assist the teachers in expositive classes and students as a reference material. This material presents a process designed in BPMN notation (Weske, 2012), addressing all knowledge areas for the initiating and planning processes groups. Thus, this process defines the correct sequence to use dotProject+ functionalities, satisfying the requirements to execute each process activity before initiating it. Also, it presents print screens for each process activity, explaining how to use dotProject+ functionalities to support that part of the process (Figure 4).

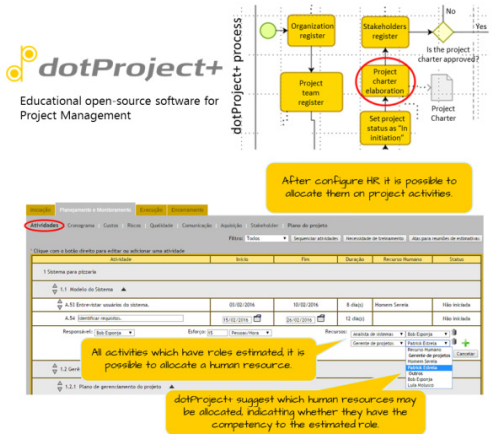


Figure 4: dotProject+ - Usage Guide.

All instructional materials are freely available and may be downloaded from dotProject+ web site (http://www.gqs.ufsc.br/evolution-of-dotproject/). Thus, any teacher interested may download all material and then apply the IU.

[1] This evaluation process has been approved by CEPSH/UFSC – an ethic committee for researches with human beings, and is registered under the number - 47734215.9.0000.0121.

4.2 IU Evaluation Process

The IU evaluation aims to identify its quality in relation to its content, instructional materials, user experience, students learning, and instructional strategy, based on students and teachers perspectives. It is carried out by a series of case studies, based on the empiric study process defined by Wohlin (2012) (Figure 5)[1]. This evaluation process is integrated with the GQM approach (Basili et al., 1994), which is utilized to define the evaluation goal, the analysis questions which evaluate this goal, and metrics which support answering these questions.

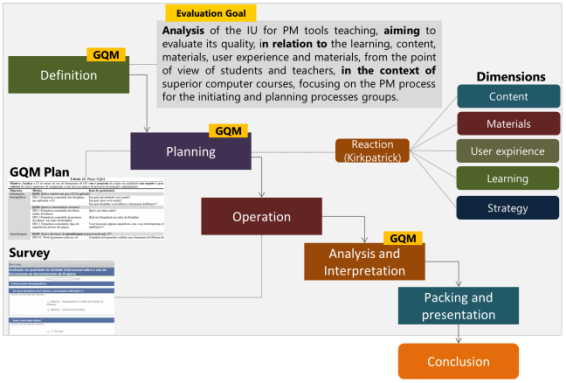


Figure 5: IU evaluation process (Wohlin et al., 2012; Basile et al., 1994).

Derived from the GQM metrics, data collection instruments were developed. There is a questionnary for stundents and other for teachers, both containing the same structure: demographic questions followed by a set of affirmations using a likert scale to evaluate their perception of each IU dimension. These dimensions were chossen based on its compatibility with the proposed IU, and in accordance with previous studies (Arcuri and Fraser, 2012; Chen et al., 2013). At the end, there are open questions to collect points regarding strengths and improvements to be made, as well as other comments. The individual participation in the IU evaluation is voluntary and anonymous.

The perception about the IU quality is also evaluated by observation, analyzing the students and teachers behavior when interacting with the instructional materials, and when carrying out the instructional activities. In this case the data is collected by verbal or written feedback, that may be provided by e-mail or using an online form that was available for students and teachers along all the IU application.

5 CASE STUDY DEFINITION

Aiming to analyze the IU for teaching PM tools using dotProject+, a case study has been defined (Figure 6). The case study definition presents all steps necessary for this study to be reproduced in future situations (Wohlin et al., 2012).

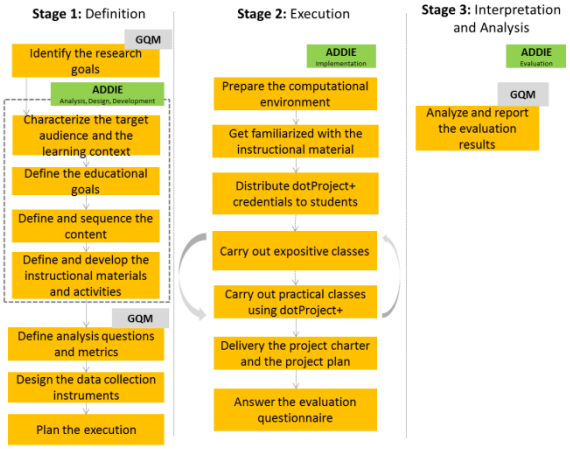


Figure 6: Case study definition (Wohlin et al., 2012; Basile et al., 1994).

The case study is organized in 3 stages. The first one is related to its definition, where the research goals are defined, and the IU is developed, addressing the analysis, design and development phase of ADDIE process. In this phase the IU evaluation is also defined, and the data collection instruments are developed. Details about the follow stages are presented in the next section.

6 CASE STUDIES EXECUTION

The case study has been reproduced during six consecutive semesters. It has been used in 3 different Brazilian educational institutions, applied by 6 different teachers, in a total of 13 classes, teaching more than 300 students (Table 2).

These case studies execution begins with teacher preparing the computational environment, installing dotProject+ and creating student accounts. The teacher also has to get familiarized with the instructional plan, with dotProject+ usage and with the usage guide.

At the beginning of classes, the students are organized in groups, receiving their credentials for

Table 2: Cases studies execution.

Semester	Educational institution	Course	Discipline	Teacher*	Number of students
2013-1	UFSC	Computer Science	Planning and management of software projects	Teacher A	19
2013-2	UFSC	Computer Science	Planning and management of software projects	Teacher A	21
2013-2	UFSC	Information Systems	Project management	Teacher B	23
2014-1	UFSC	Computer Science	Planning and management of software projects	Teacher A	30
2014-1	UFSC	Information Systems	Project management	Teacher B	22
2014-1	SEBRAE – Espirito Santo	Environmental Technician	Project management	Teacher C	19
2014-2	UFSC	Computer Science	Planning and management of software projects	Teacher A	17
2015-1	UFSC	Computer Science	Planning and management of software projects	Teacher A	24
2015-1	UFSC	Information Systems	Project management	Teacher D	19
2015-1	SENAC – Jaraguá do Sul	Information Technology Management	Fundamentals in Project Management	Teacher E	21
2015-2	SENAC – Jaraguá do Sul	Information Technology Management	Fundamentals in Project Management	Teacher E	24
2015-2	UFSC	Information Systems	Project management	Teacher D	37
2015-2	UFSC	Computer Science	Planning and management of software projects	Teacher E	28

Legend:
UFSC – Federal University of Santa Catarina.
SENAC – National Service of Commercial Learning.
SEBRAE - Brazilian Service of Assistance of Small and Medium Enterprises.
* Teacher names have been replaced for privacy reasons.

dotProject+ access, and the teacher carries out expositive classes explaining how to access dotProject+, as well as general navigation rules. During the next meetings the teacher employs the usage guide to explain how to use dotProject+ to support the PM process for one or more knowledge areas, until the students complete the whole PM process for the initiating and planning processes groups. At the end of the classes, the students may export the project charter and the project plan in PDF format, and it is delivered to the teacher for evaluation.

After the IU application, the students and teachers are invited to answer an evaluation questionnaire. The answer of this questionnaire is non-mandatory, anonymous and online. Once the data are collected, it is analyzed and discussed, identifying the IU quality for each dimension, and its strengths and improvement points.

It is important to highlight that the case studies were not exactly reproduced, because the IU materials have been improved each semester, based on the feedback we have received, both by students and teachers. This feedback is normally related to improvement suggestions or from reporting some implementation issue. The feedback was provided verbally or in writing, using an online form or e-mail. In regards to data collection instruments, we have developed and applied the complete evaluation questionnaire (derivated from GQM), but just in the case studies carried out after the second semester of 2014. In the previous semesters we applied a questionnaire with open questions for individuals to inform their improvement suggestions for the IU and its perceived strengths, as well as other general comments.

7 ANALYSIS

In this section we present the collected data from the students perspective, based on the received answers of the evaluation questionnaire. The analysis is segmented by each evaluation dimension, presenting the median of each questionnaire item, considering the 26 students that have answered it.

Firstly, concerning the **content dimension** (Figure 7), among the affirmations there are items about the content relevance for computer professionals, and whether it is addressed in proper depth and extension.

The **materials dimension** aims to evaluate the students perception about dotProject+ and its usage guide. The dotProject+ (Figure 8) was evaluated

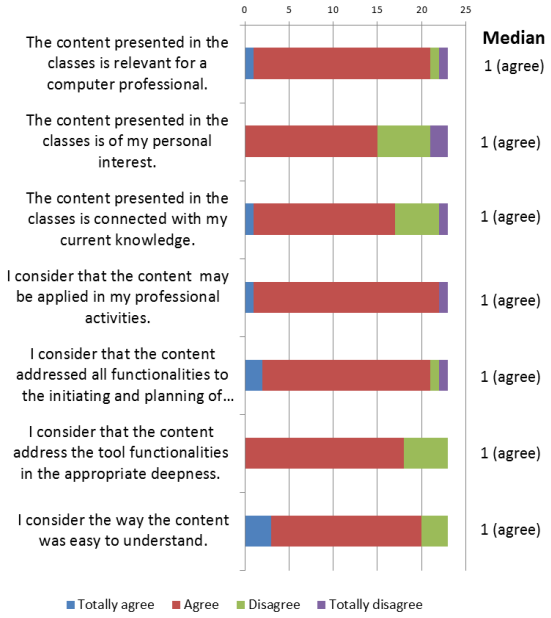


Figure 7: Content dimension evaluation data.

based on affirmations related to its contribution to the understanding about the practical application of the PM process, and also if it also assisted students during the instructional activities.

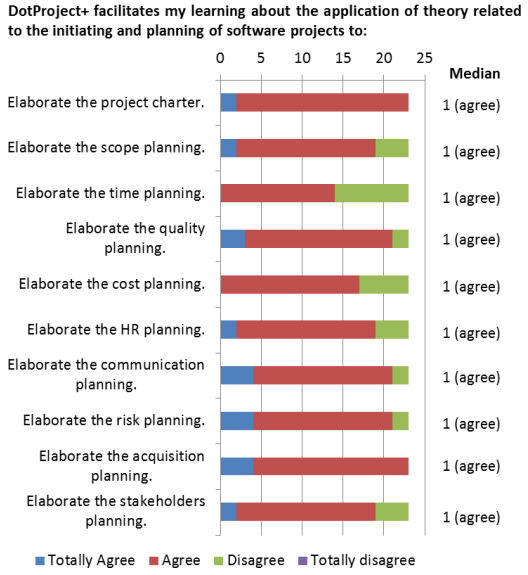


Figure 8: dotProject+ evaluation data for PM process coverage.

Yet, related to dotProject+, it was collected data about the students perception of its usability (Figure 9), and how much it stimulate students and the dificults they may had faced during its usage.

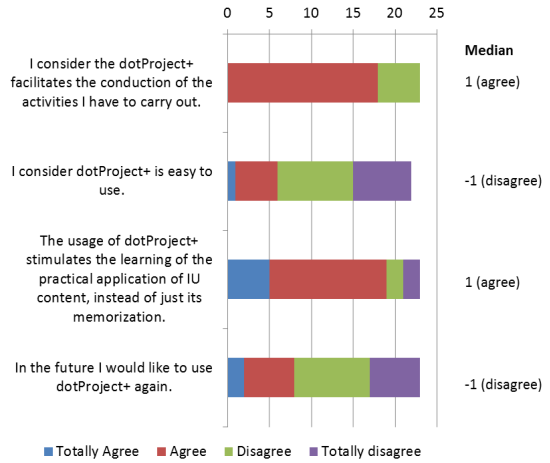


Figure 9: dotProject+ usability evaluation data.

Furthermore, we collected data to evaluate how the usage guide has contributed to students learning (Figure 10); identifying how it was consulted during the classes, and whether its content and structure are suitable for students learning.

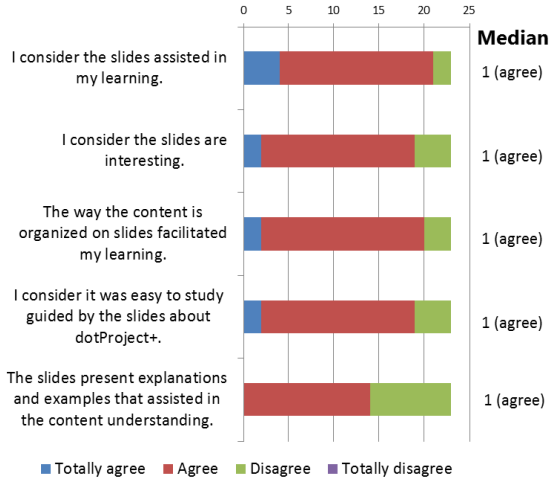


Figure 10: Usage guide slides evaluation data.

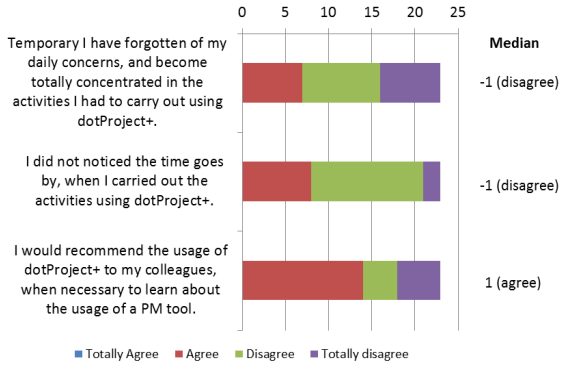


Figure 11: User experience dimension evaluation data.

The **user experience dimension** (Figure 11) was evaluated based on affirmations that attempted to identify how students become motivated when carrying out the instructional activities.

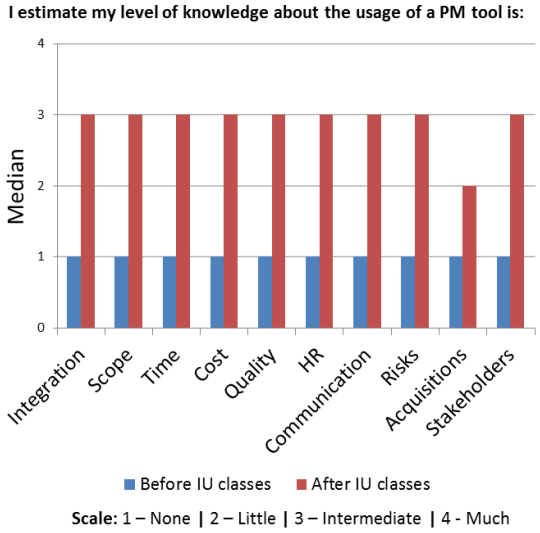


Figure 12: Learning dimension evaluation data.

Then, in relation to the **learning dimension** (Figure 12), we have utilized affirmations to understand the knowledge about PM tools usage in the beginning of the IU, and how it was after the IU.

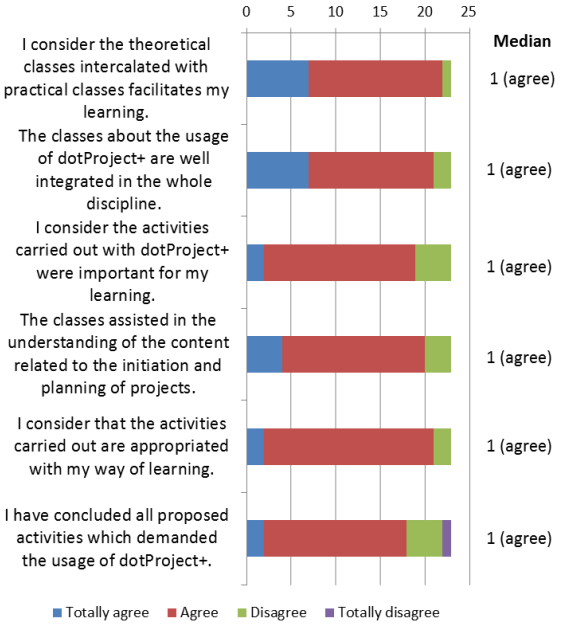


Figure 13: Instructional strategy dimension evaluation data.

The **instructional strategy dimension** (Figure 13) was evaluated based on affirmations about the contribution of theoretical classes and instructional

activities in students learning. In addition to its suitability to students learning preferences.

From the teachers' perspective, 5 of the 6 teachers have answered the IU evaluation questionnaire. Since we have more data about students evaluation than about teachers evaluation, we opted to present only the former. Nevertheless, in the discussion section we are addressing both perspectives.

8 DISCUSSION

In this section we present a discussion about the IU quality focusing on each dimension we have defined to be evaluated. We are considering the students and the teachers perspectives, collected both by the evaluation questionnaires and by observation.

Firstly, about the **learning dimension**, the students stated that the IU assists them to correlate the theoretical content with the professional practice, through the functionalities they learned to use on dotProject+. Also, the majority of students have finished all activities they were delegated, indicating they effectively have learned how to use a PM tool to elaborate a project charter and a project plan, covering all PM knowledge areas. From the teachers perspective they highlighted that the IU assisted the student learning, mainly because the employed instructional strategy and materials are strictly designed for the IU learning objectives.

Regarding the **instructional materials dimension**, the students stated that the usage guide assisted in their understanding about the whole PM process, explaining how to use dotProject+ functionalities to carry out each process activity. However, specifically about dotProject+, the most addressed issue was related to its usability, which in a few cases was considered complex to use. From the teachers perspective, they considered the entire instructional material, ready to use, assisted in classes preparation. They also considered the usage guide very important, especially because of the lack of experience of most students with PM tools. They also considered that the usage of dotProject+ assists in the learning of the whole discipline, because the students have the opportunity to apply the theoretical content through the tool functionalities. As regards to dotProject+ functionalities, they highlighted that it is very positive in supporting all of these processes in a single tool. It avoids the usage of several tools to cover all these functionalities, thus facilitating the integration of its results in a complete project plan generated by the tool. However, some teachers complained about the support provided by HR allocation process, because it demands many steps, making it complex to be used. Other teachers also complained about the complexity for the installation of dotProject+, including many add-on modules, a complication when the teacher does not master the related technologies.

The issue about dotProject+ usability was drastically reduced after the 2015 first semester, when the dotProject+ version 2.0 was adopted, which had its usability improved based on an analysis carried out by a software usability researcher. In this same version we also have included a new theme, to make dotProject+ more attractive (in relation to the standard dotProject theme), also assisting in usability issues and facilitating student receptiveness. Regarding the issue related to the HR allocation process, it also has been improved in the dotProject+ version 2.0, which has simplified this process.

About the **content dimension**, the IU has received a positive feedback about the content coverage and depth, both by students and teachers. It was because dotProject+ supports several functionalities, enabling to apply on practice many PM techniques that are taught in theoretical classes, covering all PM knowledge areas. Most students also considered the content relevant for a computer professional.

In relation to the **user experience dimension**, the students highlighted that the most motivational aspect of the IU is that it enables them to have a clear comprehension about the practical application of the PM content. Although their motivation had been affected by the usability issues of dotProject+, this negativity has been reduced after the dotProject+ version 2.0 was adopted. From the teachers perspective they also considered that the IU prepares the students for their professional career, and all teachers have informed that they would recommend the IU to other teachers that need to teach about PM tools.

In relation to the **instructional strategy dimension**, the students considered that the intercalation between the theoretical and the practical classes has facilitated the content understanding, and also the comprehension of its practical applicability. Regarding the teachers perspective, they highlighted that the instructional activities guided by the PM process facilitated student understanding about the correct order to use dotProject+ functionalities. Thus, the result being, that most students concluded the elaboration of the project charter and the project plan.

Based on the presented discussion, it is evident that the proposed IU makes a positive contribution to student learning, and meets for teacher demands for all evaluated dimensions. During its application several improvement suggestions have been

collected, and many of them have already been implemented. Thus, at the current stage the IU reached a maturity level that allows it to be adopted by other teachers that need to teach about PM tools, aligned with the PM process as defined by PMBOK.

In comparison to related studies, thought the evaluation of the presented IU, it has demonstrated to assist in the teaching of a more extensive part of PM process than any other related studies. However, some studies, such as Gregoriou et al. (2010), besides covering just time and human resources knowledge areas for planning processes group, it offers several specific instructional features to assist students. Among these features are the configuration of difficulty level, automatic feedback, and tutorial videos. On the other hand, only the presented IU has adopted a material as the usage guide, which is oriented by the PM process, and provides instructions about how each step of this process may be supported by PM tools functionalities.

8.1 Threats to Validity

As any research there are some threats to validity (Wohlin et al., 2012). They are analyzed for conclusion, construction, and external threats to validity.

Threats to conclusion validity may occur due to inconsistences in the data collected. In this research the individuals may lack some knowledge related to PM, even while being taught during the discipline. It may lead to wrong interpretation of questionnaire items and as a consequence lead to inconsistent answers. To reduce this threat, the questionnaire was designed carefully analyzing the employed terminology, bringing it as near as possible to the student language. Also, when the students answered the questionnaire still in the context of the discipline, they may have been afraid to be punished for their answers, especially when criticizing some IU aspects. This was mitigated by anonymising the answers, applying the questionnaire only after all student evaluations had been concluded and having this final process conducted by an external researcher, instead of the teacher. However, especially from a students perspective, a significant part of our evaluation has been based on data collected in an ad-hoc manner, based on verbal and written feedback provided during the case studies instances.

Threats to construction validity are related to the data collection instrument, which may not contain the necessary set of questions to reach the evaluation goal. We have employed the GQM approach to design the questionnaire, thus the evaluation goal was systematically deployed in question analysis and metrics, which were represented by questionnaire items.

Threats to external validity may occur by not obtaining a significant sample. In fact, we still do have not collected a significant amount of structured data; consequentially performing the statistical study with only 26 students answers. However, we have mitigated that by applying the IU in different semesters, involving 6 teachers and 304 students, which are significant events for a general evaluation of the IU for teaching PM tools with the support of dotProject+.

9 CONCLUSIONS

This study has presented the evaluation of an IU for teaching the usage of PM tools. This IU has introduced the educational PM tool - dotProject+, which is an enhancement of one of the most popular open-source tools for PM. DotProject+ includes a more comprehensive support to the PM process and educational features. After carrying out a series of case studies, and analysing the collected data from students and teachers perspectives, the IU has demonstrated to be effective for teaching the usage of PM tools for the initiating and planning processes groups, covering all PM knowledge areas. Students highlighted that they have learned the content, and consider they are able to reproduce it in their professional activities. Teachers have highlighted they would like to use this IU again, and would recommend it to other teachers. Future studies may expand upon the instructional feedback of dotProject+, beyond the creation of other IUs to address other processes groups that were not included in this research.

ACKNOWLEDGEMENTS

This work was supported by the CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico – www.cnpq.br), an entity of the Brazilian government focused on scientific and technological development.

REFERENCES

ACM, IEEE Computer Society, 2013. Computer Science Curricula 2013.

Arcuri A., Fraser G., 2012. Sound Empirical Evidence in Software Testing. In: *Proc. of 34th International Conference on Software Engineering*, Zurich/ Switzerland.

Basili, V., Caldier, G., Rombach, D., 1994. The Goal Question Metric Approach. *Encyclopedia of software engineering*, pp. 528–532.

Branch R., 2009. *Instructional Design: The ADDIE Approach.* Springer, 2nd edition.

Car Ž., Belani H., Pripužić K., 2007. Teaching Project Management in Academic ICT Environments. In*: Proc. of the Int. Conf. on computer as a tool,* Warsaw.

Cicibas H., Unal O., Demir K, 2010. A comparison of project management software tools (PMST). In: *Proc. of the 9th Software Engineering Research and Practice*, Las Vegas.

Chen H., Chen Y., Chen K., 2013. The Design and Effect of a Scaffolded Concept Mapping Strategy on Learning Performance in an Undergraduate Database Course. *IEEE Transactions on Education*, vol. 56, n. 3, pp. 300-307.

Fabac R., Radoševic D., Pihir I., 2010. Frequency of use and importance of software tools in project management practice in Croatia. In: *Proc. of 32nd Int. Conf. on Information Technology Interfaces*, Cavtat.

Gregoriou G., Kirytopoulos K., Kiriklidis C., 2010. Project Management Educational Software (ProMES). *Computer Applications in Engineering Education*, vol. 21, n. 1, pp. 46–59.

Gonçalves R., Wangenheim C., 2015. How to Teach the Usage of Project Management Tools in Computer Courses: A Systematic Literature Review. In: *Proc. of the Int. Conf. on Software Engineering and Knowledge Engineering*, Pittsburgh.

Hill, H., Rowan, B., Ball, D., 2005. Effects of Teachers' Mathematical Knowledge for Teaching on Student Achievement. *American Educational Research Journal*, 42(2), p. 371-406.

Keil, M., Rai, A., Mann J., 2003. Why software projects escalate: The importance of project management constructs. *IEEE Transactions on Engineering Management*, vol. 50, n.3, pp. 251–261.

Kirkpatrick, D., kirkpatrick, J., 2012. *Evaluating Training Programs: The Four Levels*. Berrett-Koehler Publishers, 4nd edition.

Mishra A., Mishra D., 2013. Software Project Management Tools: A Brief Comparative View, *ACM SIGSOFT Software Engineering Notes*, 38 (3), pp. 1-4.

Ojeda O., Reusch P., 2013. Sustainable procurement - Extending project procurement concepts and processes based on PMBOK. In: *Proc. of 7th International Conference on Intelligent Data Acquisition and Advanced Computing Systems*, Berlin/Germany, pp. 530 – 536.

Pereira, A., Gonçalves R., Wangenheim, C., 2013. Comparison of open source tools for project management. *International Journal of Software Engineering and Knowledge Engineering*, vol. 23, n. 2, 2013, pp. 189-209.

PMI – *Project Management Institute, 2013. A Guide to the Project Management Body of Knowledge*, 5. ed., Newtown Square.

Reid K., Wilson G., 2007 DrProject: A Software Project Management Portal to Meet Educational Needs. In: *Proc. of the Special Interest Group on Computer Science Education*, Covington.

Salas-Morera L., Arauzo-Azofra A., García-Hernández, L., 2013. PpcProject: An educational tool for software project management*. Computers & Education,* vol. 69, n. 1, pp. 181-188.

The Standish Group, 2013. Chaos Manifesto 2013, Boston.

Wohlin C., Runeson P., Höst M., 2012. *Experimentation in Software Engineering: An Introduction*, Springer.

Weske, M., 2012. *Business Process Management: Concepts, Languages, Architectures*. Springer, 2nd edition.

OvERVIeW: Ownership Visualization Word Cloud

Ilenia Fronza and Stefano Trebeschi
Free University of Bozen-Bolzano, Piazza Domenicani, 3, 39100, Bolzano, Italy
ilenia.fronza@unibz.it, stefano.trebeschi@gmail.com

Keywords: OvERVIeW, Word Cloud, Visualization, Code Ownership, SVN.

Abstract: In many situations, awareness about code ownership is important; for example, this awareness might allow to contact the responsible person(s) of a piece of code to get clarifications. Source versioning systems can provide information about code ownership. However, the considerable amount of data collected might prolong the time to retrieve the information needed. OvERVIeW addresses this issue: it collects data from a versioning system and visualizes developers' effort using a well-known and intuitive visualization, the word cloud. We applied pre-attentive processing principles in the designing phase, which use graphical properties (e.g., form and color) that are processed pre-attentively, i.e., they are understood faster. In our visualization, each word represents a class; the number of lines added and removed (during a given time period) is used as size metric, and the color represents the developer(s) working on the code. We show how OvERVIeW can be used to visualize three different cases of code ownership: collective, weak, and strong. We report a sample application of OvERVIeW in the context of a multi-developer OSS project.

1 INTRODUCTION

The work of many developers is required to create almost any non-trivial piece of code. Large teams face communication and coordination issues, and splitting software development across a distributed team make it even harder to achieve an integrated product (Herbsleb and Grinter, 1999). Open Source (OSS) projects represent the typical situation where coordination problems arise, since developers contribute from around the world, meet face-to-face infrequently, and need to coordinate virtually (Crowston et al., 2005). For example, the entry barrier is a problem that has been acknowledged by OSS developers (Cubranic and Booth, 1999): a newcomer needs to understand the existing code and read the available documentation. Usually, this is a very time consuming and tedious task. In this case, information about code ownership would allow contacting directly the responsible person(s) of a piece of code in order to get explanations. To this end, source versioning systems can provide information about code ownership. However, the considerable amount of data collected might prolong the time to retrieve the information needed.

The key to solve these issues, and to promote coordination in general, is increasing the level of awareness; in particular, information of *who* is changing *what* in the system (i.e., code ownership) has been proposed as a means to increase awareness in the team (Lanza et al., 2010). To this end, a large number of visualizations has been proposed in Software Engineering, mostly to show the evolution of software systems (Caudwell, 2010; Pinzger et al., 2005; Voinea et al., 2005; Wettel et al., 2011; Ciani et al., 2015), or development effort and authors (D'Ambros et al., 2005; Lanza et al., 2010; Ogawa and Ma, 2008; Vervloesem, 2010). Visualizations are often taken without any adjustments from other disciplines, for which they were specifically designed, and they do not convey information quickly and effectively. The problem is relevant as, if properly designed, visualizations can help people getting information effectively (Few, 2012; Fronza, 2013). On the contrary, bad-designed ones can be confusing.

In this paper, we propose a tool, OvERVIeW, that provides to the developer an overall understanding of code ownership in a project. OvERVIeW uses a well-known and intuitive visualization, the word cloud (Feinberg, 2010), where: a) each word represents a class; b) the number of lines added and removed (during a given time frame) is used as size metric; and c) the color represents the developer(s) working on the code. We used pre-attentive processing principles (Ware, 2012), so that the observer can get quickly the information needed. We propose three different code ownership cases (Fowler, 2006): a) collective, when each class has been developed by many developers;

Fronza, I. and Trebeschi, S.
OvERVIeW: Ownership Visualization Word Cloud.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 405-412
ISBN: 978-989-758-187-8

b) weak, when some parts of the system are developed only by one developer; and c) strong, when each developer is responsible of a part of the system. A sample application of OvERVIeW, in the context of a OSS multi-developer project, shows concretely its potentials.

The paper is organised as follows: Section 2 discusses existing works in this area; Section 3 details the design rationale; Section 4 shows the usage of OvERVIeW in the context of a multi-developer OSS project; Section 5 draws conclusions and future work.

2 RELATED WORK

In this Section we provide an overview of the existing visualizations of development effort; then, we introduce the main principles of data visualization, word clouds, and their existing applications.

2.1 Visualizing Development Effort

Several visualizations have been proposed to show development effort. An overview is provided in (Tornhill, 2015), where effort visualizations are proposed in order to evaluate knowledge drain in a codebase, or to learn social pitfalls of team work. A spider chart is used in (Diehl, 2007) to show various software metrics, including effort. RUP hump charts are used in (Heijstek and Chaudron, 2008) to depict effort distribution through the development process. Fractal figures are used in (D'Ambros et al., 2005) to show whether many people contributed to the development of a class and the intensity of each contribution.

Many tools have been proposed to support the exploration of code structure, change histories, and social relationships, as well as to animate a projects history, such as:

- ProjectWatcher mines changes to the source code and provides a graph where the color of packages and classes indicates who edited the class most recently (Schneider et al., 2004);
- CVSscan shows, at the source code level, how code changes during the development process (Voinea et al., 2005);
- StatSVN uses SVN repositories to generate statistics regarding the development process, both for the overall project and for individual authors (Jason and Gunter, 2006);
- StarGate groups developers in clusters corresponding to the areas of the file repository they work on the most (Ogawa and Ma, 2008);
- Code Swarm shows the history of commits in a OSS project as a video (Ogawa and Ma, 2009);
- Gource displays the logs from a version control system as an animated tree (Caudwell, 2010);
- UrbanIt provides a visualization of software repositories together with evolutionary analyses (Ciani et al., 2015).

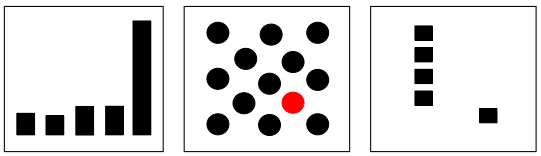
Still, many of the available visualizations are technical and some time is needed to understand them correctly, especially by non-experts in information visualization.

2.2 Principles of Data Visualization

Several studies have demonstrated the importance of pre-attentive processing, which is the unconscious accumulation of information from the environment. In other words, the brain can process the available information and filter the relevant message (Few, 2012). A visualization is *well-designed* when it is able to induce the viewer's brain to memorize only the information that should be communicated. According to (Ware, 2012) and (Few, 2013), information visualization should consider the following pre-attentive properties to maximize the understanding of the information, to guide attention, and to enhance learning:

- form (i.e., length, width, orientation, shape, size and enclosure), which is widely applied in data visualization. The bar chart, for example, pre-attentively shows data using the length;
- color (and intensity), which is applied to saturation and lightness;
- spatial position, which is the perception of the dimensional space, in terms of differences in vertical and in horizontal positions of elements.

Figure 1 shows some examples of these principles.



(a) Form (b) Color (c) Position

Figure 1: Examples of pre-attentive processing principles.

2.3 Word Clouds

Word clouds are graphical representations of word frequency that present a picture of the most common words used, with those used more often displayed larger. Words are placed in the playing field (i.e., the space that can be filled by words) which is

often a cloud, although plenty of shapes are available. Word clouds are usually colorful, but colors are meaningless. The key point of this visualization is its understandability even by non-experts (Feinberg, 2010). For this reason, word clouds have been applied in a range of fields to provide a quick summary of texts pulled from, for example, websites and blogs for different purposes, including sentiment analysis and market research.

In Software Engineering, word clouds have been used, for example, for requirements engineering. In fact, since word clouds are understandable both by experts and non-experts, they are thought to be an excellent communication means between customers and developers. For example, (Delft et al., 2010) shows an application of word clouds that is built over an automatic analysis of spreadsheets and text documents. The user can split and organize concepts, which are represented by words, in entities. The resulting visualization is a set of word clouds which are graphically and semantically interconnected. Word clouds have been also applied to improve the development process; Lanza et al. (2010) propose the usage of word clouds to inform the developer if somebody else has recently changed one of the classes she/he is currently working on. The general purpose of this application of word clouds is to promote the coordination of developers working on the same piece of code. To the best of our knowledge, the usage of word clouds to visualize code ownership is novel in the field.

Despite their wide usage in many disciplines, the following aspects of word clouds are considered controversial: 1) colors are meaningless, 2) the context is lost, and 3) random orientation of words makes difficult to read the graph (Feinberg, 2010; Harris, 2010; Cui et al., 2010).

3 Designing OvERVIeW

This Section introduces a usage scenario and describes the design rationale.

3.1 Scenario

Sepp is a programmer and S is a OSS project that he has just joined. To start his activities, Sepp has been asked to fix some bugs in S. Now, suppose (not an off chance) S to be almost undocumented, unstructured, and poorly written. Sepp starts running the program, examining the source code, and reading any available documentation. Then, Sepp uses some tools, such as source code browsers and static analysers. Sepp is downhearted, and he decides to ask the developers for some explanations. To this end, he needs to know who is responsible of what. Sepp needs to get this information as quickly as possible, otherwise he will probably give up and leave the project.

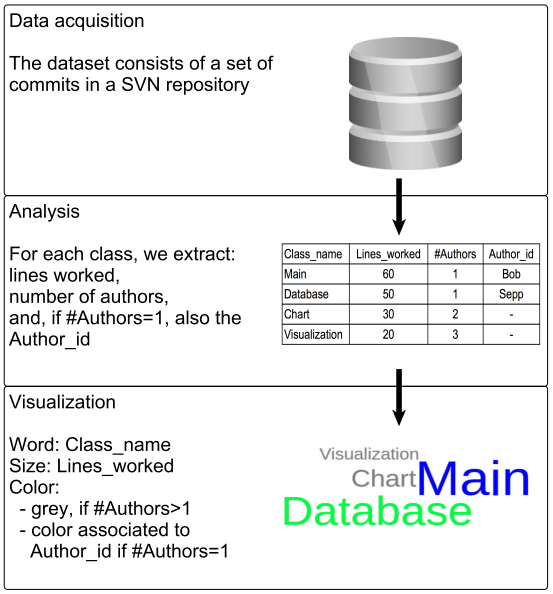


Figure 2: Schema of the proposed approach.

3.2 Design Rationale

To overcome this or similar scenarios, we created OvERVIeW by addressing the three main steps of the visualization pipeline reported in (Diehl, 2007): data acquisition, analysis, and visualization. Figure 2 shows a schematic view of the proposed approach.

3.2.1 Data Acquisition

OvERVIeW extracts a set of commits from a SVN repository and, from each commit, one or more unique atomic changes. A change is composed of the fully qualified name of the class, authors of the commit, the number of lines added, the number of lines removed, and the timestamp of the commit.

3.2.2 Analysis

The set of changes is restricted to a particular time frame (e.g., one month) and grouped by class name. Afterwards, OvERVIeW extracts, for each class:

- *Class Name*, as the name of the class without its path;
- *Lines Worked* (L), as the sum of lines added and lines removed from the class in each change (Moser et al., 2008): $L = \sum L_{add} + \sum L_{del}$;

- *Number of Authors* (n_{auth}), as the number of distinct authors that have worked on the class;
- *Author IDs*, as the ID(s) of the author(s).

3.2.3 Visualization

OvERVIeW generates a word cloud, where each word is a class name and carries the properties of size and color. *Size* is defined as $size = f(L)$, where L are the lines worked, and f is a positive, discrete, monotonically increasing function. We tested for linear, logarithmic, exponential and square root functions (Feinberg, 2010). The latter provided the most appreciable results. *Color* is mapped to a categorical colormap T for a single-developer code, and to grey (i.e., (0.5, 0.5, 0.5) in RGB) otherwise. When a class is associated with multiple authors (i.e., it is represented by a grey color), OvERVIeW stores the list of authors. To visualize the names of the developers, the user can click on the class and a message box is superimposed (Figure 3). To be considered as an author of a class, a developer must satisfy a minimum threshold in terms of lines of code developed; this threshold depends on the type of project, the team and other environmental factors.

Figure 3: Message box showing the developers of a class.

The layout of a visualization influences its perception (Few, 2012; Lohmann et al., 2009). To obtain an effective design, we addressed the controversial aspects of word clouds (Section 2.3) as follows.

Context. Word clouds have been originally created for text analysis. Word size represents the frequency of words occurrences. This way, concepts and themes are completely lost (Harris, 2010; Cui et al., 2010). Moreover, stop words[1] are simply removed from the visualization, and the meaning of the text can be modified by this removal. In OvERVIeW, this problem is solved a priori, since there are no stop words to be removed, and no themes or context proper of a natural language.

Shape. In word clouds, words are placed in the playing field through a randomised greedy algorithm. Once the word is placed, its position does not change and the algorithm checks if that word overlaps another word or crosses the boundaries of the playing field. If the playing field is too small, most of the words will fall outside the field. If, on the contrary, the playing field is too large, the shape will be an incoherent blob, as every non-intersecting position will be acceptable. As a general rule, the playing field must be large enough to contain at least the largest word (Feinberg, 2010). In OvERVIeW, the size of the playing field is fixed. The vector of words sizes is scaled by a factor α to fit the largest word in the field. Thus, the proportion among sizes is maintained and the playing field is fitted. Words in the middle of the playing field attract more user attention than those near the borders (Lohmann et al., 2009; Bateman et al., 2008). In our case, the observer should focus on bigger words, as they might represent core classes. Therefore, OvERVIeW places bigger words in the center of the playing field.

Orientation. We choose for all the words an horizontal placement, which is the best choice in terms of readability (Few, 2012).

Colors. Usually, in word clouds colors are meaningless and are used for "aesthetic appeal" (Feinberg, 2010). In OvERVIeW, colors depend on the author(s) of the class in the given time frame.

To guide the design of OvERVIeW, we used graphical properties that are processed pre-attentively, meaning that they are understood faster (Ware, 2012). Pre-attentive elements have been grouped into: form, color, motion, spatial position. We decided to use the following properties in OvERVIeW:

- *Form:* classes that required more effort (i.e., having higher L) are bigger, as the observer should notice first the parts of the project absorbing most of the effort, and larger words have been shown to attract more attention than smaller ones (Lohmann et al., 2009);
- *Spatial Position:* larger words are in the center (Lohmann et al., 2009), as the observer needs to focus on the parts of the project absorbing most of the effort;
- *Color:* different types of code ownership can easily be recognized using the colors in the word cloud (Section 3.3).

The following Section explains how OvERVIeW can be used to show different types of code ownership.

3.3 OvERVIeW in Action

OvERVIeW allows to reason about code ownership in a software project. Figure 4 shows an example. The number of lines added and removed (during the given

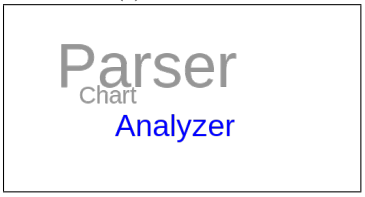
[1]Stop words are words that appear frequently in a natural language (e.g., "it", "do", "not"). They do not have a meaning themselves, but they assume a meaning when associated to another word.

time frame) is used as size metric; each color corresponds to a developer, and a word (i.e., a class) is grey when more than one developer has worked on it during the given time frame. Thus, Figure 4 shows that the distribution of effort among the classes does not change in the three cases, as the dimension of words does not change. The change of colors in the three cases, instead, allows to recognize the following three cases of code ownership:

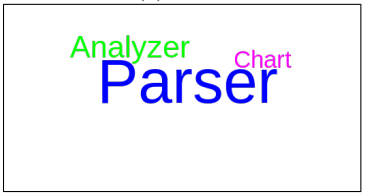
1. Collective. Figure 4(a) is completely grey, meaning that each class has been developed by many developers.
2. Weak. Figure 4(b) is mostly grey. Most of the system has been developed by multiple developers, but part of the code (i.e., "Analyzer") is *owned* only by developer 1, because it is blue;
3. Strong. In Figure 4(c) each developer is responsible of a part of the system: developer 1 works only on "Parser", developer 2 works only on "Analyzer", and developer 3 works only on "Chart".

The three types of code ownership are depicted in Figure 4.

(a) Collective



(b) Weak



(c) Strong

Figure 4: Sample application of OvERVIeW to show the three cases of code ownership. The distribution of effort among the classes does not change in the three cases. The change of colors allows to recognize three cases of code ownership.

4 CASE STUDY

In order to show concretely the potentials of OvERVIeW, we applied it to a OSS project, Epsilon (http://www.eclipse.org/epsilon/), a framework of the Eclipse project, which aims at providing consistent and interoperable languages for MDE tasks. We retrieved data from the SVN repository of the Epsilon project from December 2011 to December 2012. In December 2011 the project had 2340629 lines of code; in December 2012 it reached 2385455 lines of code. Therefore, 44826 lines were added in one year of activity. In the same period, 3 developers committed at least once.

We applied OvERVIeW with time frames of 15 days. Figure 5 shows the OvERVIeW word clouds of different time periods during the analysed year. In particular, Figure 5(a) shows that only green-developer and pink-developer were working on the project during the first two weeks of December 2011. The names of the classes they are working on suggest their responsibilities. Words like "parser", "token", "lexer", and "generation" are green; therefore, we can suppose the green-developer to be responsible for compilation aspects. This seems to be confirmed in Figure 5(d) where "AbstractParser" is also green. Debug analysis and development seem to be done mostly by the blue-developer, as all the classes related to these activities are blue in Figure 5(c). Moreover, in all the word clouds, there are just a few grey classes, meaning that developers tend not to work on the same parts of code.

Overall, each developer in Epsilon seems to have a specific responsibility with respect to a subsystem. Therefore, this project seems to have a strong code ownership, in particular if we consider that the four word clouds have been selected from a time frame of one year. This excludes the possibility of finding the developers devoted to one particular task, thus working on a specific part of the code in that period.

4.1 Qualitative Evaluation

In order to be "useful", a visualization should convey information in an understandable, effective, easy-to-remember way. Evaluation is needed to assess the usefulness of a visualization. Two main types of evaluation exist (Diehl, 2007):

- *quantitative* methods measure properties of the visualization, of the algorithm, or of the human observer interacting with the visualization. Quantitative evaluation requires a statistical analysis of the results of a controlled experiment;
- *qualitative* methods gather data about the individual experience of human observers with the visualization. When it comes to the human perception of and interaction with a visualization, qualitative

(a) 1-15 December 2011

(b) 5-20 April 2012

(c) 14-28 June 2012

(d) 6-20 September 2012

Figure 5: Example using Epsilon.

evaluation methods are of the outmost importance for various reasons, including the fact that they cover more aspects of the visualization. In particular, "task-oriented" analysis is very popular, because it provides an answer to a key question from a user's point of view: "Does the system solve the user's problem?"

In this work we apply qualitative, task-oriented evaluation to answer the following question from a user's point of view: "Does OvERVIeW show effectively the information about code ownership?". To this end, we contacted the community of Epsilon to have a confirmation (or negation) of the information that we got by looking at the output of OvERVIeW; this means that we simulated to be a user in front of the word cloud generated by OvERVIeW. All the developers confirmed that each of them tends to focus on certain parts of the code, due to research interests at the time; anyway, they pretty much share some knowledge of the code. In particular, the green-developer is the founder of the project; he has written most of the code, and he has done improvements to ease the grammar building, and the other two members of the community have contributed to it in the past. The green-developer and the blue-developer did the first iteration of the debugging facilities; the latter has added some functionality, such as inspecting variables and doing step-by-step execution.

5 CONCLUSIONS AND FUTURE WORK

This paper describes OvERVIeW, a tool to collect data from a versioning system and to visualize code ownership using a well-known and intuitive visualization, the word cloud. We explained how OvERVIeW can be used to visualize three different cases of code ownership: 1) collective, 2) weak, and 3) strong. We described a sample application of OvERVIeW in the context of a multi-developer OSS project.

Altogether, we think that OvERVIeW is valuable and worth further investigation. Therefore, as future work we plan to improve the capabilities of OvERVIeW. For example, in this work the "owner" of a class is the developer who has been the only one working on it in the selected time frame. Our definition of "ownership" should be improved to consider possible collaborations and pair programming activities, which are suggested during knowledge transfer to improve productivity and quality (Fronza and Succi, 2009; Fronza et al., 2011b; di Bella et al., 2012). Moreover, we will deal with the scalability of our approach. To this end, we plan to improve OvERVIeW as follows:

- OvERVIeW should be able to deal with long and mostly meaningless class names, sometimes occurring in projects. Indeed, the current solution, which ignores the path name for class names, might not handle overloaded class names. Moreover, longer names might also receive more user attention than shorter ones with a similar size metric, because they occupy more space.

- In the case study shown in this paper, only a limited number of developers is participating to the project. Visualizing data of projects with large teams might result in too colorful graphs. We plan to consider bigger communities and to be able to focus on sub-communities in our visualization.
- The trade-offs between stability and visual clutter should be investigated more formally. In order to improve the readability of the word clouds, in the case study of this paper a short period (i.e., two weeks) was selected, as the team presented a high level of activity. OvERVIeW should be able to deal with high levels of activity of the teams.

The word cloud is generated at one point in time. This steady approach can be expanded by including time information. Time information can be represented, for example, by the time elapsed since the last commit, or the time during which a developer did not commit at all, or the time in which a code is handled only by a developer. This would require a multivariate facet approach, in which it is not sufficient to use a simple transfer function to map only intensity information. The combination of the resulting tool with a prediction algorithm (Abrahamsson et al., 2011; Fronza et al., 2011a) would enable to visualize, e.g., the evolution of effort distribution in the project.

Furthermore, interactive techniques for flexible word cloud navigation and manipulation should be considered. For example, the technique used in (Liu et al., 2014) supports multifaceted viewing of word clouds .

In this work we applied qualitative, task-oriented evaluation to understand if OvERVIeW shows information effectively and we received positive feedback; still, evaluation needs to be extended. In particular, we need to assess if the output of OvERVIeW is understandable and easy-to-remember. To this end, we plan to perform an experiment to evaluate OvERVIeW by asking developers to perform some tasks using different visualizations (including OvERVIeW) and to provide their feedback about OvERVIeW. Finally, we plan to perform case studies using more OSS projects. In this context, it would be useful to collect feedback from users that are living a scenario such as the one described in Section 3.1.

REFERENCES

Abrahamsson, P., Fronza, I., Moser, R., Vlasenko, J., and Pedrycz, W. (2011). Predicting development effort from user stories. In *Empirical Software Engineering and Measurement (ESEM), 2011 International Symposium on*, pages 400–403.

Bateman, S., Gutwin, C., and Nacenta, M. (2008). Seeing things in the clouds: the effect of visual features on tag cloud selections. In *Proceedings of the nineteenth ACM conference on Hypertext and hypermedia*, HT '08, pages 193–202, New York, NY, USA. ACM.

Caudwell, A. H. (2010). Gource: visualizing software version control history. In *Proceedings of the ACM international conference companion on Object oriented programming systems languages and applications companion*.

Ciani, A., Minelli, R., Mocci, A., and Lanza, M. (2015). Urbanit: Visualizing repositories everywhere. In *Software Maintenance and Evolution (ICSME), 2015 IEEE International Conference on*, pages 324–326.

Crowston, K., Wei, K., Li, Q., Eseryel, U. Y., and Howison, J. (2005). Coordination of free/libre open source software development. In *In Proceedings of the International Conference on Information Systems (ICIS 2005), Las Vegas*, pages 181–193.

Cubranic, D. and Booth, K. S. (1999). Coordinating open-source software development. In *Proceedings of the 8th Workshop on Enabling Technologies on Infrastructure for Collaborative Enterprises*, WETICE '99, pages 61–68, Washington, DC, USA. IEEE Computer Society.

Cui, W., Wu, Y., Liu, S., Wei, F., Zhou, M., and Qu, H. (2010). Context-preserving, dynamic word cloud visualization. *Computer Graphics and Applications, IEEE*, 30(6):42 –53.

D'Ambros, M., Lanza, M., and Gall, H. (2005). Fractal figures: Visualizing development effort for CVS entities. In *Proc. Int'l Workshop on Visualizing Software for Understanding (Vissoft)*, pages 46–51. IEEE Computer Society Press.

Delft, F., Delft, M., and van Deursen Delft, A. (2010). Improving the requirements process by visualizing end-user documents as tag clouds. In *Proc. of Flexitools 2010*.

di Bella, E., Fronza, I., Phaphoom, N., Sillitti, A., Succi, G., and Vlasenko, J. (2012). Pair programming and software defects - a large, industrial case study. *IEEE Transactions on Software Engineering*, 99(PrePrints):1.

Diehl, S. (2007). *Software Visualization: Visualizing the Structure, Behaviour, and Evolution of Software*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Feinberg, J. (2010). Wordle. In Steele, J. and Iliinsky, N., editors, *Beautiful Visualization: Looking at Data through the Eyes of Experts*, chapter 3. O'Reilly Media.

Few, S. (2012). *Show me the numbers : designing tables and graphs to enlighten*. Analytics Press.

Few, S. (2013). *Information Dashboard Design: Displaying data for at-a-glance monitoring*. Analytics Press.

Fowler, M. (2006). Code ownership. Retrieved Feb. 10, 2016, from http://martinfowler.com/bliki/CodeOwnership.html.

Fronza, I. (2013). Opening statement. *Cutter IT Journal*, 26(1):3–5.

Fronza, I., Sillitti, A., Succi, G., and Vlasenko, J. (2011a). Failure prediction based on log files using the cox proportional hazard model. In *SEKE*, pages 456–461. Knowledge Systems Institute Graduate School.

Fronza, I., Sillitti, A., Succi, G., and Vlasenko, J. (2011b). Understanding how novices are integrated in a team analysing their tool usage. In *Proceedings of the 2011 International Conference on Software and Systems Process*, ICSSP '11, pages 204–207, New York, NY, USA. ACM.

Fronza, I. and Succi, G. (2009). Modeling spontaneous pair programming when new developers join a team. In Abrahamsson, P., Marchesi, M., and Maurer, F., editors, *Proceedings of XP2009, Pula, Sardinia, Italy, May 25-29, 2009.*, volume 31 of *Lecture Notes in Business Information Processing*, pages 242–244. Springer.

Harris, J. (2010). Word clouds considered harmful. Nieman Journalism Lab. Retrieved Feb. 10, 2016, from http://www.niemanlab.org/2011/10/word-clouds-considered-harmful/.

Heijstek, W. and Chaudron, M. (2008). Evaluating rup software development processes through visualization of effort distribution. In *Software Engineering and Advanced Applications, 2008. SEAA '08. 34th Euromicro Conference*, pages 266 –273.

Herbsleb, J. D. and Grinter, R. E. (1999). Splitting the organization and integrating the code: Conway's law revisited. In *Proceedings of the 21st international conference on Software engineering*, ICSE '99, pages 85–95, New York, NY, USA. ACM.

Jason, K. and Gunter, M. (2006). Statsvn: Statistics for svn repositories based on the open source project statcvs. in CSI5140, Winter 2006, Available from: http://www.statsvn.org/.

Lanza, M., Hattori, L., and Guzzi, A. (2010). Supporting collaboration awareness with real-time visualization of development activity. In *In Proceedings of CSMR 2010 (14th IEEE European Conference on Software Maintenance and Reengineering*. IEEE CS Press.

Liu, X., Shen, H.-W., and Hu, Y. (2014). Supporting multifaceted viewing of word clouds with focus+context display. *Information Visualization*.

Lohmann, S., Ziegler, J., and Tetzlaff, L. (2009). Comparison of tag cloud layouts: Task-related performance and visual exploration. In *Proceedings of the 12th IFIP TC 13 International Conference on Human-Computer Interaction: Part I*, INTERACT '09, pages 392–404, Berlin, Heidelberg. Springer-Verlag.

Moser, R., Pedrycz, W., and Succi, G. (2008). A comparative analysis of the efficiency of change metrics and static code attributes for defect prediction. In *Proceedings of the 30th international conference on Software engineering*, ICSE '08, pages 181–190, New York, NY, USA. ACM.

Ogawa, M. and Ma, K.-L. (2008). Stargate: A unified, interactive visualization of software projects. In *Visualization Symposium, 2008. PacificVIS '08. IEEE Pacific*, pages 191 –198.

Ogawa, M. and Ma, K.-L. (2009). code_swarm: A design study in organic software visualization. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1097–1104.

Pinzger, M., Gall, H., Fischer, M., and Lanza, M. (2005). Visualizing multiple evolution metrics. In *Proceedings of the 2005 ACM symposium on Software visualization*, SoftVis '05, pages 67–75, New York, NY, USA. ACM.

Schneider, K. A., Gutwin, C., Penner, R., and Paquette, D. (2004). Mining a software developers local interaction history. In *In Proceedings of the International Workshop on Mining Software Repositories*, pages 106–110.

Tornhill, A. (2015). *Your Code as a Crime Scene: Use Forensic Techniques to Arrest Defects, Bottlenecks, and Bad Design in Your Programs*. Pragmatic Bookshelf.

Vervloesem, K. (2010). Visualizing open source projects and communities. LWN.net. Linux info from the source. Retrieved Feb. 5, 2016, from https://lwn.net/Articles/382468/.

Voinea, L., Telea, A., and van Wijk, J. J. (2005). Cvsscan: visualization of code evolution. In *Proceedings of the 2005 ACM symposium on Software visualization*, SoftVis '05, pages 47–56, New York, NY, USA. ACM.

Ware, C. (2012). *Information Visualization: Perception for Design*. Interactive Technologies. Elsevier Science.

Wettel, R., Lanza, M., and Robbes, R. (2011). Software systems as cities: a controlled experiment. In *Proceedings of the 33rd International Conference on Software Engineering*, ICSE '11, pages 551–560, New York, NY, USA. ACM.

Software Evolution of Legacy Systems
A Case Study of Soft-migration

Andreas Fürnweger, Martin Auer and Stefan Biffl
Vienna University of Technology, Inst. of Software Technology and Interactive Systems, Vienna, Austria
{*martin.auer, stefan.biffl*}*@tuwien.ac.at*

Keywords: Software Evolution, Migration, Legacy Systems.

Abstract: Software ages. It does so in relation to surrounding software components: as those are updated and modernized, static software becomes evermore outdated relative to them. Such legacy systems are either tried to be kept alive, or they are updated themselves, e.g., by re-factoring or porting—they evolve. Both approaches carry risks as well as maintenance cost profiles. In this paper, we give an overview of software evolution types and drivers; we outline costs and benefits of various evolution approaches; and we present tools and frameworks to facilitate so-called "soft" migration approaches. Finally, we describe a case study of an actual platform migration, along with pitfalls and lessons learned. This paper thus aims to give software practitioners—both resource-allocating managers and choice-weighing engineers—a general framework with which to tackle software evolution and a specific evolution case study in a frequently-encountered Java-based setup.

1 INTRODUCTION

Software development is still a fast-changing environment, driven by new and evolving hardware, operating systems, frameworks, programming languages, and user interfaces. While this seemingly constant drive for modernization offers many benefits, it also requires dealing with legacy software that—while working—slowly falls out of step with the surrounding components that are being updated—for example, if a certain version of an operating system is no longer supported by its vendor. There are various ways to handle such "aging" software: one can try to keep it up and running; to carefully refactor it to various degrees to make it blend in better; to port its code; to rewrite it from scratch. The main stakeholders in deciding on a course of action are managers, which must allocate resources to and consider the risks and maintenance cost profiles of the various options (e.g., will affordable developers with specific skills still be available?), as well as software developers, which should be aware of the long-term implications of their choices (e.g., will a certain programming language be around in five years' time?).

To provide some software evolution guidelines, our paper first gives an overview on software evolution types, covering maintenance, reengineering, and whether to preserve or redesign legacy systems. We address software aging and its connection with maintainability. We look into different aspects of software maintenance and show that the classic meaning of maintenance as some final development phase after software delivery is outdated—instead, it is best seen as an ongoing effort. We also discuss program portability with a specific focus on porting source code.

We then outline costs and benefits of various evolution approaches. These approaches are either legacy-based, essentially trying to preserve as much as possible of the existing system, or migration-based, where the software is transferred, to various degrees, into a new setup.

After that, we focus on various methods for "soft" migration approaches—those approaches aim to facilitate traditional migration methods like porting or rewriting code via support tools and frameworks. We especially concentrate on the Java programming language and present a specific variant of a soft-migration approach, which is using a Java-based program core with several platform-specific branches.

Finally, we describe a case study of an actual soft migration of the UML editor UMLet, which is currently available as a Swing-based Java program and an SWT-based Eclipse plugin, and which is ported to a web platform. We analyze some problems we encountered, and discuss the benefits and drawbacks of the suggested approach.

Fürnweger, A., Auer, M. and Biffl, S.
Software Evolution of Legacy Systems - A Case Study of Soft-migration.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 413-424
ISBN: 978-989-758-187-8

2 RELATED WORK

(Mens and Demeyer, 2008) give an overview of trends in software evolution research and address the evolution of other software artifacts like databases, software design, and architectures. A general overview of the related topics of maintenance and legacy software is given by (Bennett and Rajlich, 2000), who also identify key problems and potential solution strategies.

Lehman classifies programs in terms of software evolution and also formulates laws of software evolution (Lehman, 1980; Lehman et al., 1997), which are, however, not considered universally valid (Herraiz et al., 2013). There are also many exploratory studies that try to analyze and understand software evolution based on specific software projects (Johari and Kaur, 2011; Businge et al., 2010; Zhang et al., 2013; Ratzinger et al., 2007; Kim et al., 2011). (Chaikalis and Chatzigeorgiou, 2015) develop a prediction model for software evolution and evaluate it against several open-source projects. (Benomar et al., 2015) present a technology to identify software evolution phases based on commits and releases.

The related topic of legacy systems is a bit ambiguous, due to differing definitions. It can describe a system that resists modification (Brodie and Stonebraker, 1995), a system without tests (Feathers, 2004), or even all software as soon as it has been written (Hunt and Thomas, 1999). A natural question regarding legacy systems is whether to preserve or redesign them. As this question is not easy to answer (Schneidewind and Ebert, 1998), the pros and cons of reengineering or preserving a system are to be compared thoroughly before making a decision (Sneed, 1995). In addition, it is possible to replace a system in stages to minimize the operational disruption of the system (Schneidewind and Ebert, 1998).

Even though the classic view of maintenance as the final life-cycle phase of software after delivery is still prevalent, it is a much broader topic, especially for programs which must constantly adapt to a changing environment. There are reports that the total maintenance costs are at least 40% of the initial development costs (Brooks Jr., 1995), 70% of the software budget (Harrison and Cook, 1990), and up to 90% of the total costs of the system (Rashid et al., 2009). As these numbers show, the topic of maintenance is crucial. (Lientz and Swanson, 1980) categorize maintenance activities into distinct classes. Several authors (Sjøberg et al., 2012; Riaz et al., 2009) propose maintainability metrics.

Finally, when migrating a system, a reengineering phase is almost always necessary. According to (Feathers, 2004), this phase should be accompanied by extensive testing to make sure the application behavior stays the same. (Fowler and Beck, 1999) list useful refactoring patterns, while (Feathers, 2004) stresses how legacy code can be made testable.

3 SOFTWARE EVOLUTION

This section outlines relevant disciplines and nomenclature related to software evolution.

3.1 Overview

(Lehman et al., 2000) divide the view on software evolution into two disciplines. The *scientific discipline* investigates the nature of software evolution and its properties, while the *engineering discipline* focuses on the practical aspects like *"theories, abstractions, languages, activities, methods and tools required to effectively and reliably evolve a software system."* (Lehman, 1980) classifies programs based on their relationship to the environment where they are executed. Lehman also formulates the eight *laws of software evolution*. Among those laws, two aspects are emphasized: *continuing change* (i.e., without adaption, software can become progressively less effective), and *increasing complexity* (i.e., as software evolves, its complexity tends to increase unless effort is spent to avoid that). According to (Herraiz et al., 2013), these laws have been proven in many cases, but they are not universally valid.

3.2 Legacy Systems

There are different definitions of what a legacy system or legacy code is. (Brodie and Stonebraker, 1995) describe it as *"a system which significantly resists modification and evolution."* (Feathers, 2004) defines it as code without tests, while (Hunt and Thomas, 1999) state that *"All software becomes legacy as soon as it's written."*

Preserve or Redesign Legacy Systems

According to (Schneidewind and Ebert, 1998), the question whether to preserve or redesign a legacy system is not easy to answer. In general, most organizations do not rush to replace legacy systems, because the successful operation of these systems is vital. But they must eventually take some action to update or replace their systems, otherwise they will not be able to take advantage of new hardware, operating systems, or applications.

An important aspect of this decision is that one does not have to choose an extreme solution like preserving a system unaltered, or redesigning it from scratch. Instead, the existing system can be maintained while the replacement system is developed, which makes a fluid transition from the old to the new system possible. This minimizes the disruption to the existing system and avoids replacing the existing system as a whole while it is operational (Schneidewind and Ebert, 1998).

(Sneed, 1995) remarks that reengineering is only one of many solutions to the typical maintenance problems with legacy systems. He also mentions that there must be a significant benefit, like cost reduction or added value, to justify the reengineering, and that it is important to compare the maintenance costs of the existing solution to the expected improvements introduced by the reengineering.

3.3 Software Aging

"Programs, like people, get old. We can't prevent aging, but we can understand its causes, take steps to limits its effects . . . and prepare for the day when the software is no longer viable." (Parnas, 1994)

The maintenance costs of an aged application tend to increase, because modifications to a software generally make future adaptions more difficult. Therefore it is important to invest time to keep software modules simple, to clean up convoluted code, and to redesign program logic if necessary (Monden et al., 2000).

3.4 Maintenance

Software maintenance is sometimes considered to be the final phase of the delivery life-cycle. Unfortunately, this definition is outdated for many types of software, which must constantly adapt to changing requirements and circumstances in their environment.

Maintenance Effort and Costs

In large software codebases, the required maintenance effort is high. (Basili et al., 1996) show how to build a predictive effort model for software maintenance releases, with the goal of getting a better understanding of maintenance effort and costs. (Brooks Jr., 1995) claims that the total maintenance costs of a widely used program are typically at least 40% of the initial development costs. (Rashid et al., 2009) show that over the last few decades the costs of software maintenance have increased from 35-40% to over 90% of the total costs of the system. According to (Harrison and Cook, 1990), more than 70% of the software budget is spent on maintenance; 75% of software professionals are involved with maintenance. According to (Coleman et al., 1994), HP has between 40 and 50 million lines of code under maintenance, and 60% to 80% of research and development personnel are involved in maintenance activities.

Maintenance Classes

(Lientz and Swanson, 1980) categorize maintenance activities into four classes: *adaptive* (keeping up with changes in the software environment); *perfective* (new functional or nonfunctional user requirements); *corrective* (fixing errors); and *preventive* (prevent future problems). The most maintenance effort (around 51%) falls into the second category, while the first category (around 23%), and the third one (around 21%), make up most of the remaining effort.

There are several metrics to evaluate how maintainable a system is. Unfortunately, these methods don't always produce consistent results (Sjøberg et al., 2012; Riaz et al., 2009). (Sjøberg et al., 2012) consider the overall system size to be the best predictor of maintainability.

3.5 Reengineering

"Reengineering (. . .) is the examination and alteration of a subject system to reconstitute it in a new form and the subsequent implementation of the new form. Reengineering generally includes some form of reverse engineering (to achieve a more abstract description) followed by some form of forward engineering or restructuring." (Chikofsky and Cross II, 1990)

Many times the existing software is a legacy system, although *"it is not age that turns a piece of software into a legacy system, but the rate at which it has been developed and adapted without having been reengineered."* (Demeyer et al., 2002)

(Feathers, 2004) mentions that in the case of legacy systems the necessary reengineering phase has to be more elaborate and should be accompanied by the introduction of automated tests, to make sure the current application behaves the same before and after the reengineering. (Gottschalk et al., 2012) describe reengineering efforts to reduce the energy consumption of mobile devices.

3.6 Portability of Programs

Older high-level languages like C always aimed to be portable across systems, but often fall short, e.g., due to different APIs or system word size. To solve these problems, new languages were designed that run on

virtual machines. This was a huge step forward in terms of portability, as programs are compiled into an intermediate language that is runnable without modifications on any system with an implementation of the required virtual machine. Java is an example of such a language.

A similar approach is taken by web applications, which require a web browser instead of a virtual machine. The browser-based approach has other advantages like easy distribution. Web applications run on every platform with modern browser. Newer approaches based on system virtualization and containers (like Docker) address the need for better portability of whole subsystems without any restrictions on programming language or the used ecosystem.

Java Language and Platforms

Java is a programming language specifically designed for portability, achieved via virtual machines. They cover nearly all platforms, from smart cards and mobile phones to desktop and server environments. The most familiar non-official platform is probably Android, which supports large portions of the JavaSE API excluding graphical related portions such as Swing and AWT.

Other uses of the language are based on compilation of Java code to another programming language, such as GWT (Google Web Toolkit), which compiles from Java to JavaScript, or J2ObjC, which compiles from Java to Objective-C. Although most transpilers support a large part of the source language's features and API, certain features cannot be mapped to the target language (e.g., classes that are used for the Java GUI Framework Swing are not supported in GWT).

The main advantage of such a source-to-source compiler (also known as *transpiler*) is that there is no need for a Java Virtual Machine. This is especially important for the web platform, because even though browser-plugin-based Java Applets are possible, the plugin is based on the Netscape plugin API (NPAPI), which is not supported by mobile browsers. Furthermore, desktop browsers have also started to remove NPAPI support, e.g., the Chrome browser removed it on September 1, 2015.[1]

4 COSTS AND BENEFITS

This section discusses software evolution types and costs and benefits of migration/preservation.

[1] support.google.com/chrome/answer/6213033

4.1 Types of Software Evolution

Simplified, software evolution comes in various flavors (in increasing order of perceived costs), and is characterized by the following activities:

Legacy-based Evolution

1. Simple maintenance
   - Keep the system running.
   - Only apply bugfixes and required changes.
2. Maintenance with some reengineering
   - Carefully adapt and overhaul program logic.
   - Document application logic.
   - Create automated tests if missing.

Migration-based Evolution

3. Soft migration
   - Use tools to ease migration (e.g., virtual machines, transpilers, ...).
   - Reuse as much as possible the core parts of the legacy source.
   - Only add minimal code in new languages (e.g., Java wrapper around existing COBOL application; HTML pages for GWT transpiled code).
4. Hard migration or porting
   - Re-program the application from scratch.
   - Re-compile existing code on new target platform.

At first glance, the costs seem to increase in this list of evolutionary steps. However, this need not be the case:

- As for (1), legacy systems set up with old programming languages (ADA, COBOL) might incur increasing maintenance costs due to a lack of available expertise.
- With respect to (4), well-programmed C-code, on the other hand, can theoretically be ported to, i.e., re-compiled on, a new operating system at almost zero cost. (In practice, this very rarely happens; even supposedly platform-independent languages like Java often cause portability problems.)

4.2 Software Evolution Criteria

After outlining some terminology and various aspects of software evolution, we can now summarize costs, risks, and benefits involved in migrating software to help determine the appropriate software evolution type.

Table 1: Comparison of costs/risks and benefits of preservation.

Preservation Risks	Preservation Benefits
Legacy systems are hard to maintain and change.	Stability (training, operations, ...) is preserved.
Underlying, external dependencies (e.g., hardware, operating systems, virtual machines, software frameworks) could become difficult or impossible to obtain, risking an inability to operate the software.	Better predictability of overall system costs (if no major changes are required).
User acceptance for the software might wane, and the user base might erode, as users flock to other vendors with more modern approaches, like updated GUIs, or solutions running on new systems. For example, endusers might choose to use windows-based GUIs over their command-line-based ancestors.	Saved resources can be applied to keep the software alive with minor, and less dangerous, software evolution steps than outright migration, like partial reengineering, documentation via reverse-engineering, or virtualization.
If the software components, languages, or frameworks are becoming obsolete, it might get more difficult and/or costlier to find the required programming expertise (witness the numerous COBOL systems still running in insurance and banking). Maintenance efforts and costs will likely increase over time.	

Table 2: Comparison of costs/risks and benefits of migration.

Migration Risks	Migration Benefits
Obviously, setting up or re-writing software is expensive and the costs are often difficult to estimate. The original software's long-developed optimizations and workarounds might not always be easy to reproduce with completely new technology.	Modern languages and related tools, a larger programmer base, faster hardware, ..., can reduce costs of new feature development, maintenance, and error fixing.
Choosing new environments, setups, and languages as migration target carries the risk of selecting wrong candidates, like soon-to-be obsolete OSes or language paradigms. New, buzz-word-rich platforms often fade and disappear quite unceremoniously.	Modern new software frameworks and libraries can improve the user experience, maintainability, and testability of the system.
There are considerable risks of introducing bugs or unwanted software behavior. Even seemingly useful bug fixes can lead to problems, e.g., if other systems, aware of the known bug, already compensate for it.	Better APIs can increase interoperability with modern software.
If parts of the system are not migrated, or if the old software needs to be kept alive (e.g., due to contractual obligations), duplicate code bases need to be maintained, and changes propagated to both.	New platforms (mobile, web, ...) can open up new markets and increase user acceptance.
Domain experts and the developers of the legacy system are probably not available anymore, therefore it can be hard to understand and re-implement the software correctly.	New code can be made more modular using object oriented design patterns, increasing its re-usability, and introduce automated tests (unit tests, integration tests, ...).
If the old system is not documented properly, knowledge that exists only implicitly within the program logic can get lost.	Vendor and platform dependency can be reduced (e.g., by removing libraries).

5 SOFT MIGRATION

Tools and frameworks can greatly facilitate software migrations; they allow for what we dub "soft" migrations. The next subsection gives a general overview on the variety of such migration assistance; the following one focuses on Java-based support.

5.1 Soft Migration Overview

System Virtual Machines

System VMs (also called Full Virtualization VMs) virtualize the complete operating system to emulate the underlying architecture required by a program. Examples are VirtualBox or VMWare.

Application Virtual Machines

Application VMs (also called Process VMs) run as a normal application inside an existing operating system. They abstract away (most) platform and operating system differences, and therefore allow the creation of platform-independent programs that can be executed using this VM. Examples are the Java Virtual Machine, the Android Runtime (ART), or the Common Language Runtime (used by the .NET Framework).

Integrated Virtual Machines

Integrated VMs can be seen as a subtype of Application VMs, because they are integrated and run within another program (e.g., as a plugin). One popular example are Java Applets, where the JVM is either part of a browser, or added with a browser plugin. Today, they are not very common anymore, because browsers started to remove the support for such plugins for security reasons (see section 3.6 about the removal of NPAPI support in browsers).

Transpilers

A transpiler is a source-to-source compiler. It compiles or translates one language to another and therefore enables code reuse between different programming languages. Examples are GWT, which transpiles from Java to JavaScript, or J2ObjC, which transpiles from Java to Objective-C.

Delegates/Wrappers

Delegates or wrappers are tools that allow interaction between system and programming language boundaries. There are several reasons to create a wrapper (like security, or usage of a different programming language), but the basic idea is to hide the underlying program and instead provide a suitable interface for the user. Examples are libraries that allow to call from COBOL to Java[2], or from Java to .NET.[3]

[2] supportline.microfocus.com/documentation/books/nx40/dijint.htm

[3] www.ikvm.net

Distribution Utilities/Platforms

These are tools to facilitate the installing and updating of applications. One example is Java Web Start, which is basically a protocol for a standardized way to distribute Java applications and their updates. Other examples are digital distribution platforms like Google Play Store or the Apple App Store.

5.2 Java-based Soft Migration

This section describes soft-migration approaches in the context of the Java platform in more detail. Java has several properties that make it a good example for software migration: it is designed for platform independence, which facilitates, e.g., mere migrations to new operating systems; it is very popular and thus there exist a wide variety of support tools; and several of its language features make concurrent support of different platforms easier than with other programming languages.

Idea

As mentioned, the Java Programming language can be run on nearly all commonly used platforms (any platform with a Java Virtual Machine (JVM) support, like Android, iOS, and GWT via transpilation). Unfortunately, not the full Java API is available on all of these platforms—therefore core Java code that is to be run on various platforms needs to be more restrictive in terms of API usage than the rest of the code.

The idea of reusing program logic on several platforms and programs, even if they do not use the same programming language, is not new. Most client/server applications already hide their internally used programming language(s) by providing a standardized type of API (e.g., CORBA, JAX-WS, or REST). This enables several programs to reuse certain functionality as if it were part of their own application code.

This soft-migration approach also encapsulates the shared functionality behind a specific API and allows different programs to reuse it. If these programs use different programming languages, the language barrier can be avoided by using transpilers (e.g., to JavaScript with GWT, or to Objective-C with J2Objc).

Supporting Technology

As mentioned in section 4.1, soft-migration relies on supporting tools. With Java, several such tools and frameworks are available:

- GWT is a Java to JavaScript compiler to facilitate migrations to web-based platforms.

- J2Objc is a Java to Objective-C compiler to port code to iOS.
- RoboVM is a Java ahead-of-time compiler and runtime, for iOS and OS X.

An alternative way to run Java applications within a web browser are Java Applets, but they depend on browser plugins, which are often limited in functionality for security reasons.

Steps

1. *Analyze the current application.* The first goal must be to understand the legacy system in its current form. The core concepts must be abstracted and a high-level architectural model must be created. Ideally, the system is amply documented; in practice, some reverse-engineering is often inevitable.
2. *Improve the architectural model.* To support migration, or to extract reusable core components, the high-level architectural model usually must be improved. This typically leads to improved modularization of the application and to the creation of a clearer, layered architectural model.
3. *Reengineer the application.* The next step is the implementation of the improved model. This is also typically the most complex step. Special care must be taken not to break original functionality, e.g., via—possibly newly introduced—unit tests. Documentation must be updated and/or kept in sync with the changes. Organically grown extensions and ad-hoc solutions or fixes should be ironed out. This is also an opportunity to clean up naming conventions, as well as build processes.
4. *Migrate to the new platform.* After the necessary reengineering steps are completed, the new platform specific code must be implemented. If the previous steps were successfully implemented, there should be clear interfaces to the shared codebase.
5. *Optional: remove code for old platform.* If the old platform should be dropped, its platform-specific code can be removed. This helps minimize maintenance efforts—even "dead" code causes obstacles when browsing/understanding a code base.

Creating New Software

In addition to the use case of migrating an existing application to a new platform, the idea of a shared Java core component can also be used when writing new software that should run on several platforms. Ray Cromwell gave a presentation at the GWT.create conference in January 2015 entitled *Google Inbox: Multi Platform Native Apps with GWT and J2ObjC*[4], where he explained details about how Google approached the development of their new product Inbox. He mentions that they share 60-70% of their code in a Java-based core component, which is (a) used as a Java Dependency for the Android application, (b) compiled to Objective-C (with J2Objc) for the iOS application, and (c) compiled to JavaScript (with GWT) for the web application.

Useful Tools

As mentioned, the Java platform offers many tools that help in keeping the codebase maintainable and modular. The following list presents some important categories of tools, and lists some examples.

- *Automated Tests.* Typically, legacy codebases have no automated tests, therefore it is risky to refactor such code, because any change can easily break previously working features. Therefore it is usually a good idea to write some tests before refactoring the code. A useful tool to write and execute tests for Java code is JUnit.[5] It can be combined with Mockito[6] to create simple mocks of dependencies. Combined with Powermock[7], even static fields, final classes, and private methods can be mocked for tests.

  As legacy codebases often consist of tightly coupled components, it might be necessary to break those dependencies (see section 5.2) before writing tests (e.g., a tightly coupled database connection is typically a problem for tests, but a tightly coupled utility class might not). Unfortunately, breaking those dependencies also involves code changes. (Feathers, 2004) describes this vicious cycle of avoiding bugs by making code testable through changes that can potentially introduce new bugs.
- *Dependency Injection.* This implements the principle of *Inversion of Control* for resolving the dependencies of a class. It basically means that objects do not instantiate their dependencies themselves, but get them injected either manually using the constructor, or by a dependency injection framework. Martin Fowler[8] describes the pattern

[4] drive.google.com/file/d/0B3ktS-w9vr8IS2ZwQkw3WVRVeXc
[5] junit.org
[6] code.google.com/p/mockito
[7] www.powermock.org
[8] www.martinfowler.com/articles/injection.html

in detail and compares it to some alternatives (like the Service Locator pattern).

The advantages of using dependency injection become apparent in this migration-approach, as the shared code must not depend on the platform-specific implementation of any dependency. Examples for frameworks supporting dependency injection are Google Guice[9] or Spring.[10]

- *Static Code Analysis Tools.* These tools can find potential bugs, dead or duplicate code, and they can help to enforce a common code style. Examples: FindBugs,[11] PMD,[12] or Checkstyle.[13]
- *Build and Dependency Management.* Tools like Apache Maven[14] or Gradle[15] manage the dependencies of an application and its submodules. They also standardize several other aspects of an application like the directory structure and the build process. Their "convention over configuration" approach[16] also helps to familiarize new developers with a cade base, simply because of familiar project structure conventions.

6 EXEMPLARY MIGRATION

UMLet (Auer et al., 2009; Auer et al., 2003) is a UML tool in active development since 2001. It is referenced in 200+ publications, as well as 16+ books on software engineering. UMLet is the most favored plugin on the Eclipse Marketplace (Eclipse is the world-leading Java integrated development environment). In the 12 months leading up to August 1st 2015, more than 700.000 page views to UMLet's main web site have been recorded via Google Analytics.

UMLet uses a text-based approach of customizing UML elements (e.g., entering the line *fg=red* in the elements properties text block will color the background of the element red). Text without a specific meaning is simply printed, which is, e.g., a fast way to declare class methods.

To provide an exemplary application of the suggested soft-migration approach, UMLet gets migrated to a modern GWT-based web application that runs without browser plugins, while the Swing and Eclipse plugin versions are retained.

[9] github.com/google/guice
[10] spring.io
[11] findbugs.sourceforge.net
[12] pmd.sourceforge.net
[13] checkstyle.sourceforge.net
[14] maven.apache.org
[15] www.gradle.org
[16] softwareengineering.vazexqi.com/files/pattern.html

6.1 Legacy/Migration Criteria

Section 4.2 suggests two main decision drivers with the current UMLet codebase:

- The user base might move to new, web-based platforms, e.g., yUML[17], sketchboard[18], js-sequence-diagrams[19] or websequencediagrams[20].
- The current two-level platform (Java virtual machine on top of an OS) is not very future-proof:
  - Java often does not come pre-installed; it is not unlikely that future closed-source OS iterations further discourage Java deployments.
  - OS vendors like Apple increasingly limit the installation of unsigned software, or try to coax applications to be provided via custom app stores. This gives vendors the influence to prohibit flexible, uncomplicated installs for casual users, and also allows them to ban applications outright (e.g., if an application does not comply with some user interface guidelines, if the vendor perceives its usability or uniqueness as not adequate, or if tech specs like access right handling are not to the vendor's liking).

These two criteria are the main drivers to use a migration approach with the goal of increasing the platform independence of UMLet.

6.2 Analysis

A first analysis shows that most of the applications code is tightly coupled with Swing classes. The main building blocks of the diagram (UML-Classes, -UseCases, -Relations, ...), which are called GridElement, all extend the Swing class JComponent.

The Eclipse plugin provides a small SWT-based wrapper around the Swing-based code to make it runnable in Eclipse. The parsing of the elements text is done within an overwritten JComponent.print() method, therefore there is no clear separation between parsing and drawing.

6.3 Reengineering

The goals of the UMLet reengineering are to:

- separate parsing and drawing of element properties;
- remove coupling between GridElement classes and Swing-specific classes;

[17] yuml.me/diagram/scruffy/class/draw
[18] sketchboard.me
[19] bramp.github.io/js-sequence-diagrams
[20] www.websequencediagrams.com

- introduce an abstraction layer with generic draw methods instead of directly relying on Swing Graphics objects;
- move GridElements to a separate module.

Even after the reengineering, there will be a relatively large portion of platform-specific code. E.g., the composition of the graphical user interface will still be platform-specific and must be implemented separately for GWT and Swing.

Based on this analysis, the given high-level architectural model can be retained with only minor differences on each platform (e.g., file-IO handling). The main restructuring of the model consists of a clear definition on how the properties of GridElements get parsed and drawn by each platform.

6.4 Implementation of Shared Codebase

As mentioned, the shared codebase mostly consists of the GridElements and the appropriate parsing and drawing logic.

New GridElements

The new GridElements have a unified syntax for the commands and therefore break backwards compatibility with some old diagrams. They are also reduced to a smaller set of customizable elements to avoid unnecessary element duplication.

Reusable Commands on Properties

The concept of element properties (and functions triggered by specific commands) is implemented using a separate parsing procedure, which is executed every time an element changes its properties or size. During this procedure, all possible commands for the specific element are checked and—if triggered—executed.

The main advantage of this approach is that these functions can be shared between elements. If two elements need to implement, e.g., the command *bg=red* to set the background color to red, they can refer to the same generic function. Changes like new features or bugfixes to such a function will therefore automatically be applied to all elements relying on them.

Common Drawing API

Platform-specific drawing logic is hidden behind a platform-independent API, which offers basic methods like drawLine(), drawRectangle(), printText(), as well as styling methods like setBackgroundColor() or setLineThickness(). Every platform has to implement this API and redirect the calls to the underlying graphical framework (e.g., Swing in JavaSE, or the HTML Canvas drawing methods in GWT).

Missing Basic Classes in GWT

As UMLet makes heavy use of geometric functionality, it needs classes such as Point, Line, Rectangle, ... Unfortunately, those classes are located in the AWT package and therefore not available on many platforms like GWT[21] or Android.[22] To circumvent this problem, alternative classes are created that are converted to platform-specific ones directly before drawing.

6.5 Web Implementation UMLetino

The web version of UMLet is called UMLetino and it transfers UMLet's minimalistic, text-based GUI approach to the web. The initial GUI mock was designed to look exactly like UMLet, but after further evaluation, it was apparent that a web application needs several adaptions. One difference, e.g., is the menu, which is a collapsible horizontal menu at the top border in most desktop applications, but a simple vertical menu on the left side for most web applications.

Another UI component that is different, because it is already embedded in the browser, is the tab-bar. An UMLetino-specific tab-bar below the browser tabs can be confusing and it does not prevent the user from opening multiple UMLetino tabs in the browser. It was therefore removed; users who want to work in parallel on several diagrams can rely on the native browser tabs instead.

Figure 1 shows the final, reengineered code structure in UML format.

Storing in Files or on the Web

UMLet stores diagrams in the file system. Web applications typically have limited access to it, therefore we have implemented several alternatives. Diagrams can be stored:

1. in the local storage of the browser (as a quick save/load while working on a diagram);
2. on the file system, with drag-and-drop-based import, and an export based on Data-URIs and the browser's save-as functionality;
3. on Dropbox[23] servers using the users' accounts.

[21] www.gwtproject.org/doc/latest/RefJreEmulation.html
[22] developer.android.com/reference/packages.html
[23] www.dropbox.com

The diagrams are stored using the XML-based UXF file format, which is also used in UMLet.

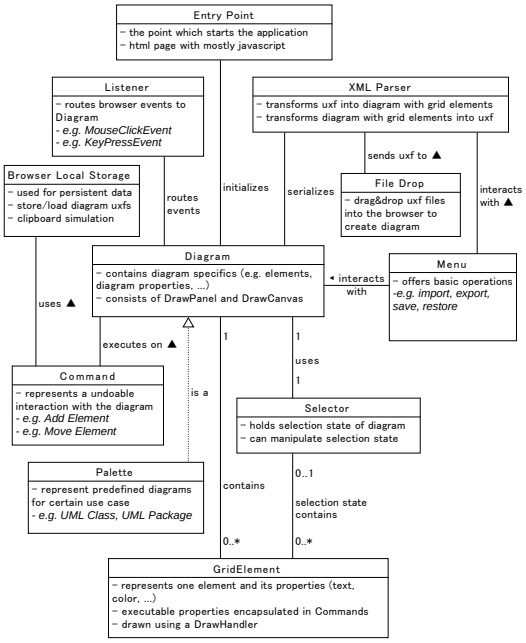


Figure 1: Reengineered code structure in UML format.

6.6 Code Base Analysis

Before Migration:

- 22,688 total (all in one project)

After Migration:

- 21,419 in Baselet (Standalone/UMLet specific)
- 8,915 in BaseletElements (shared)
- 3,135 in BaseletGWT (Web/UMLetino specific)
- 33,469 total

These numbers show that the web version consists of approx. 26% platform-specific code and 74% shared code.

The standalone version in comparison only consists of approx. 70% platform specific code and 30% shared code, but this is mostly due to the legacy support for the now deprecated OldGridElements. The old elements consist of roughly 5,600 LOC, so as soon as they are removed, approx. 36% of code will be shared.

Furthermore, there are some elements that have not been migrated to the shared codebase until now, due to their complexity (All in One Elements), or dependency on a Java Compiler during runtime (Custom Elements). They consist of roughly 4,000 LOC and will reduce the standalone specific code even more, while increasing the shared portion.

Although containing still much more specific code than the web version, the standalone project supports 3 different sub-platforms (Eclipse plugin, Swing standalone, and batch-mode) and therefore requires more code.

The overall duration of the migration was roughly 6 months; 2 developers in a remote-team setup spent an overall effort of 400 man-hours.

6.7 Lessons Learned

During UMLet's soft migration, we encountered several generic and specific issues worth mentioning:

- *Front-end code is often more platform-dependent and should be de-coupled from business logic.* There are several graphical libraries for JavaSE like AWT, Swing, or SWT. Android and GWT offer their own APIs. One possible way of avoiding this duplication is the usage of HTML (probably with some JavaScript generated by GWT), because most modern GUI frameworks can display embedded HTML+JavaScript views. In case of UMLet the code didn't have a clear separation between GUI and business logic; therefore a significant amount of time was necessary to modularize and decouple the components of the application in order to make the extraction of a shared core component possible. Fortunately, large portions of UMLet's graphical output is drawn on a Canvas where every platform offers its own implementation with only minor differences.
- *Choosing 3rd-party libraries creates dependencies and impacts the overall portability.* If a Java program should run on several platforms it must be verified that 3rd-party libraries work on all of them. In general, such libraries are only allowed to use Java classes that are supported by the platform specific API. In addition, GWT compiles Java source code to JavaScript, i.e., the library must be available as source code and not only as compiled classes.
- *Special language features like reflection and regular expressions limit portability.* GWT does not support reflection out of the box, and the default Java RegEx classes are only partially supported. Complex Regular Expressions must use GWT specific classes that work more like JavaScript RegEx than Java RegEx. In general, if a specific JVM feature like bytecode generation or just in time compilation is used, it has to be verified if it is supported by the target platform and the used transpiler.

- *The documentation and tool support of GWT is very good, but the future is uncertain.* GWT is well documented and an Eclipse plugin eases development and testing. The GWT Dev Mode makes debugging within the IDE very convenient. Nevertheless, GWT Dev Mode is restricted to older browser versions (e.g., Firefox 26), because current browser versions have removed some required APIs (e.g., NPAPI). GWT offers the Super Dev Mode as alternative, but the Eclipse integration is only possible by using 3rd-party plugins like SDBG[24], and is less convenient.
- *Useful web applications require modern browsers.* In general, web applications that should behave like standalone desktop applications typically require certain APIs to interact with the underlying system. This is a minor inconvenience for browsers like Chrome or Firefox, which get constantly updated, but other browsers like the Internet Explorer often lag behind. UMLetino also requires some specific HTML 5 features like the Web Storage API or the File Reader API, which are only available in Internet Explorer 10+.
- *Platforms have different constraints.* Although modern browsers offer several APIs to allow deep system integration, the web platform still has many constraints that do not exist for standalone applications. One example is the interaction with the file system. Standalone applications like UMLet have full access to the file system, but web applications have only limited access. File can be read by using the HTML 5 File Reader API, but most browsers disallow write access to the file system (only Chrome allows it to a sandboxed section of the filesystem).

  Find UMLetino at www.umletino.com.

7 CONCLUSION

Software maintenance, aging, and evolution are often considered an afterthought. We hope to emphasize with this paper that software will inevitably age, and that this will surely have a non-trivial impact on its use and cost profile over time.

Within the general evolution process, planners and programmers can use a simple framework to help reach evolution decisions. A concrete instance of one application's soft migration hopefully helps to illustrate this. This should also underline how modern tools make software migration much more feasible. Future work should especially look a the recent container-based software deployment tools, especially with regard to outside interface dependencies. Of special interest are layers that interact with persistent data storage (typically databases). Another approach worth examining concerns GUI adaptability for various screen/input environments, especially as GUIs are notoriously tricky to migrate.

Finally, these considerations should not merely be applied "down the road," though this is still useful. Instead, the foreseeable eventual software evolution should be part of any decisions made during the software's *initial* design stages. Those are often crucial in making sure the software will age gracefully—and, ideally, never die.

[24] github.com/sdbg/sdbg

REFERENCES

Auer, M., Pölz, J., and Biffl, S. (2009). End-User Development in a Graphical User Interface Setting. In *Proc. 11th Int. Conf. on Enterprise Inf. Systems (ICEIS)*.

Auer, M., Tschurtschenthaler, T., and Biffl, S. (2003). A Flyweight UML Modelling Tool for Software Development in Heterogeneous Environments. In *Proc. 29th Conf. on EUROMICRO*.

Basili, V., Briand, L., Condon, S., Kim, Y.-M., Melo, W. L., and Valett, J. D. (1996). Understanding and Predicting the Process of Software Maintenance Releases. In *Proc. 18th Int. Conf. on Software Engineering (ICSE)*.

Bennett, K. H. and Rajlich, V. T. (2000). Software Maintenance and Evolution: A Roadmap. In *Proc. Conf. on The Future of Software Engineering (ICSE)*.

Benomar, O., Abdeen, H., Sahraoui, H., Poulin, P., and Saied, M. A. (2015). Detection of Software Evolution Phases Based on Development Activities. In *Proc. 23rd IEEE Int. Conf. on Program Comprehension (ICPC)*.

Brodie, M. L. and Stonebraker, M. (1995). *Migrating Legacy Systems: Gateways, Interfaces, and the Incremental Approach*. Morgan Kaufmann.

Brooks Jr., F. P. (1995). *The Mythical Man-Month*. Addison-Wesley.

Businge, J., Serebrenik, A., Brand, M. V. D., and van den Brand, M. (2010). An Empirical Study of the Evolution of Eclipse Third-party Plug-ins. In *Proc. Joint ERCIM WS on Software Evolution (EVOL) and Int. WS on Principles of Software Evolution (IWPSE)*.

Chaikalis, T. and Chatzigeorgiou, A. (2015). Forecasting Java Software Evolution Trends Employing Network Models. *IEEE Transactions on Software Engineering*, 41(6):582–602.

Chikofsky, E. J. and Cross II, J. H. (1990). Reverse Engineering and Design Recovery: A Taxonomy. *IEEE Software*, 7(1):13–17.

Coleman, D., Ash, D., Lowther, B., and Oman, P. (1994). Using Metrics to Evaluate Software System Maintainability. *IEEE Computer*, 27(8):44–49.

Demeyer, S., Ducasse, S., and Nierstrasz, O. (2002). *Object Oriented Reengineering Patterns*. Morgan Kaufmann.

Feathers, M. (2004). *Working Effectively with Legacy Code*. Prentice Hall.

Fowler, M. and Beck, K. (1999). *Refactoring: Improving the Design of Existing Code*. Addison-Wesley.

Gottschalk, M., Josefiok, M., Jelschen, J., and Winter, A. (2012). Removing Energy Code Smells with Reengineering Services. In *Beitragsband der 42. Jahrestagung der Gesellschaft für Informatik e.V. (GI)*.

Harrison, W. and Cook, C. (1990). Insights on Improving the Maintenance Process Through Software Measurement. In *Proc. Int. Conf. on Software Maintenance (ICSME)*.

Herraiz, I., Rodriguez, D., Robles, G., and Gonzalez-Barahona, J. M. (2013). The Evolution of the Laws of Software Evolution: A Discussion Based on a Systematic Literature Review. *ACM Computing Surveys*, 46(2):1–28.

Hunt, A. and Thomas, D. (1999). *The Pragmatic Programmer: From Journeyman to Master*. Addison-Wesley.

Johari, K. and Kaur, A. (2011). Effect of Software Evolution on Software Metrics. *ACM SIGSOFT Software Engineering Notes*, 36(5):1–8.

Kim, M., Cai, D., and Kim, S. (2011). An Empirical Investigation into the Role of API-Level Refactorings during Software Evolution. In *Proc. 33rd Int. Conf. on Software Engineering (ICSE)*.

Lehman, M. M. (1980). Programs, Life Cycles, and Laws of Software Evolution. In *Proc. IEEE*.

Lehman, M. M., Ramil, J. F., and Kahen, G. (2000). Evolution as a Noun and Evolution as a Verb. In *Proc. WS on Software and Organisation Co-evolution (SOCE)*.

Lehman, M. M., Ramil, J. F., Wernick, P. D., Perry, D. E., and Turski, W. M. (1997). Metrics and Laws of Software Evolution - The Nineties View. In *Proc. 4th Int. Symposium on Software Metrics (METRICS)*.

Lientz, B. P. and Swanson, E. B. (1980). *Software Maintenance Management*. Addison-Wesley.

Mens, T. and Demeyer, S. (2008). *Software Evolution*. Springer.

Monden, A., Sato, S.-i., Matsumoto, K.-i., and Inoue, K. (2000). Modeling and Analysis of Software Aging Process. In *Product Focused Software Process Improvement SE - 15*, volume 1840 of *Lecture Notes in Computer Science*, pages 140–153. Springer.

Parnas, D. L. (1994). Software Aging. In *Proc. 16th Int. Conf. on Software Engineering (ICSE)*.

Rashid, A., Wang, W. Y. C., and Dorner, D. (2009). Gauging the Differences between Expectation and Systems Support: the Managerial Approach of Adaptive and Perfective Software Maintenance. In *Proc. 4th Int. Conf. on Cooperation and Promotion of Inf. Resources in Science and Techn. (COINFO)*.

Ratzinger, J., Sigmund, T., Vorburger, P., and Gall, H. (2007). Mining Software Evolution to Predict Refactoring. In *Proc. 1st Int. Symposium on Empirical Software Engineering and Measurement (ESEM)*.

Riaz, M., Mendes, E., and Tempero, E. (2009). A Systematic Review of Software Maintainability Prediction and Metrics. In *Proc. 3rd Int. Symp. on Empirical Software Engineering and Measurement (ESEM)*.

Schneidewind, N. F. and Ebert, C. (1998). Preserve or Redesign Legacy Systems. *IEEE Software*, 15(4):14–17.

Sjøberg, D. I. K., Anda, B., and Mockus, A. (2012). Questioning Software Maintenance Metrics: A Comparative Case Study. In *Proc. 6th Int. Symposium on Empirical Software Engineering and Measurement (ESEM)*.

Sneed, H. M. (1995). Planning the Reengineering of Legacy Systems. *IEEE Software*, 12(1):24–34.

Zhang, J., Sagar, S., and Shihab, E. (2013). The Evolution of Mobile Apps: An Exploratory Study. In *Proc. Int. WS on Software Development Lifecycle for Mobile (DeMobile)*.

On the Development of Strategic Games based on a Semiotic Analysis: A Case Study of an Optimized Tic-Tac-Toe

César Villacís, Walter Fuertes, Mónica Santillán, Hernán Aules, Ana Tacuri, Margarita Zambrano and Edgar Salguero
Computer Siences Department, Universidad de las Fuerzas Armadas - ESPE, Sangolquí, Ecuador
{cjvillacis, wmfuertes, mlsantillan, hmaules, agtacuri, mezambrano, elsalguero}@espe.edu.ec

Keywords: Semiotics, Tic-Tac-Toe, Videogames, Artificial Intelligence, Semiotic Models for Videogames.

Abstract: A picture can express something instead of having a thousand words that cannot do it. This phrase, which symbolically connotes a whole scheme of a signs system, is known as Semiotics. This paper presents the process of an educational video game development based on semiotic analysis. We used Extreme Programing Agile Methodology combined with a proposal of the modified Elemental Tetrad Game Design Model to develop a video game known as "Tic-Tac-Toe". The mathematical model was implemented with Artificial Intelligence algorithms and a graphical user interface including Semiotics; this was optimized for producing an enjoyable and interactive environment. With the purpose of stimulating cognitive development of children, this research combines theories about stimulating cognitive development of children; game design model, Semiotic Analysis harnesses the Model of Aleferenko, and uses algorithms based on heuristics and numerical methods in client-server architecture. The concept was tested with a representative sample of seven to eleven years old children. The results demonstrated that educational video games with Semiotics stimulate the cognitive development of children.

1 INTRODUCTION

Aware of the importance of psychomotor activity and its impact on stimulating thought; teachers and early childhood specialists value motor activities and games. One of the most fundamental resources available for educators is educational video games. Therefore, researchers are permanently exploring new learning strategies to encourage children through educational video games. Nowadays, semiotic domains are emerging more notably and potentially, could make the videogame more attractive for children. One example of this is the customisation of the avatar in the first-person-shooter video games (Gee, 2008).

Visualization is better than verbal description; this phrase symbolically connotes a whole system of signs. Its analysis or decoding is called Semiotics. According to the Oxford Advanced Learner's Dictionary, Semiotic is "a general philosophical theory of signs and symbols that deals especially with their function in both artificially constructed and natural languages and comprises syntactic, semantics, and pragmatics". It studies the phenomena and objects of significance, sign systems, and the process of senses production (Halliday, 1978).

The connection between educational video games and semiotics has been studied for three decades. The study of Myers (1991) discusses symbols within computer games and how those symbols are transformed during play. Thorne et al. (2012) describe an exploratory study of the massively multiplayer online games with a complex form of semiotic ecologies. Huber (2013) addresses the problem by proposing a model for the interpretation of videogames based on the semiotic theory of Charles S. Peirce. Ruiz et al. (2014) used videogames to help High School students to improve their understanding of numerical evaluation of algebraic expressions. Baceviciute et al. (2014) explain the convergence and hybridization process between cognitive sciences, computer science and Artificial Intelligence (AI). Kendall, 2015 analyses serious games and Semiotics separately. Those studies claim for an integration of semiotic and artificial intelligence in video games.

The present study developed a process of an educational video game based on semiotic analysis; and tested the hypothesis of an application of

Villacís, C., Fuertes, W., Santillán, M., Aules, H., Tacuri, A., Zambrano, M. and Salguero, E.
On the Development of Strategic Games based on a Semiotic Analysis: A Case Study of an Optimized Tic-Tac-Toe.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 425-432
ISBN: 978-989-758-187-8

Semiotics in the Tic-Tac-Toe videogame improves the motivation of the childhood senses and stimulates their cognitive development.

We have used the Extreme Programing Agile Methodology (XP) combined with the Elemental Tetrad Game Design Model, in order to ensure the quality of the software using Artificial Intelligence techniques in client-server architecture. The video game that has been developed and optimized is called Tic-Tac-Toe. The concept has been tested with a representative sample of seven to eleven years old children at an elementary school.

The key contributions of this study are: (1) two prototypes software developed for the Tic-Tac-Toe game, one with Semiotics; (2) a class library implemented that represents the environment and the rules of a third party using Artificial Intelligence based on heuristics and numerical methods; and (3) a game developed combining a novel mathematical model and semiotic analysis that requires algorithms of Software Engineering to optimize the Tic-Tac-Toe's ability to apply learning theories.

2 THEORETICAL FRAMEWORK

2.1 The Elemental Tetrad Game Design Model

This model is currently used for game design. According to Gibson (2014), this model uses the events of the story and states purpose of the game. It evaluates the human-computer interaction by using the game technologies. Furthermore, it separates the basic elements of a game into four sections: (1) Mechanics: the rules for interaction between the player and the game; (2) Aesthetics: describe how the game is perceived by the five senses; (3) Technology: this element covers all the underlying technology that makes the game work; (4) Story: this describes the sequence of events in the game.

2.2 Extreme Programming (XP)

XP is an agile software development methodology. It is a lightweight methodology using a set of existing software development practices in conjunction (Schneider, 2003). According to Beck (2000), the project lifecycle of XP includes the following phases: Exploration, Planning, Iterations to Release, the Product ionizing, and Maintenance. The practices taken from XP focused on software coding needed for the game.

2.3 Semiotic Analysis of Aleferenko

This model integrates several areas of the knowledge, and allows making a connotative and denotative analysis of the symbols. For Aleferenko (Tokarev, 2014) a pyramid constitutes the coalition of the Pierce's triad. It includes two new elements: Meaning/ Connotation; and Significant/ Denotation. This allows to create a pyramid organized with the following elements: (1) Sign; (2) Object/Referent; (3) Significant/ Denotation; (4) Meaning/Connotation; and Concept. The pyramid of Aleferenko integrates the bases of Semiotics. In contrast to Pierce and Frege (Wisse, 2002), Aleferenko considers outlined models and creates a new complete model, where the connotation is the agent of analysis that engenders a deep knowledge that leaves an appropriation of the receiver.

2.4 Theories to Stimulate Cognitive Development of Children

Tic-Tac-Toe stimulates cognition of children, and the game is part of the culture as it is included in common educational practices. Lev Vygotsky (1967) assigned the game into the category of instrument and socio-cultural promoter of children's mental development, and the results showed that facilitates the development of higher functions. These are acquired through interaction with the surrounding world. The approach of mediation according to Vygotsky is perceived as the presence of people, objects, and situations that interact in various socio-cultural contexts, which can be verbal, visual and physical, and can generate experiences that affect cognitive development. Vygotsky elaborated the concept known as Zone of Proximal Development Theory (Brown, 1999), explained as: (1) the distance between the actual developmental level is determined by the ability to independently solve a problem; and, (2) the level of potential development is determined by problem solving under the mediation of an adult.

Feuerstein (1991) considers that the subject's interactions with the environment can have two modalities: (1) direct exposure to stimuli; and, (2) learning experiences through mediators. As suggested by Feuerstein, it is crucial to consider that all human beings are modifiable. To be able to fulfil this condition, we should understand mediation as an intervention strategy that tries to affect the body of mediator, seeking greater efficiency in the process of information and therefore the cognitive structure. Finally, Lipman (2002), developed his educational philosophical proposal known as "critical thinking",

that is interested in forming a thought careful, orderly, prudent and reasonable. In this sense the children are be able to make judgments as part of the practice of their own learning process in which case an educational game is considered like a learning activity.

All these theoretical assumptions analysed converge in cognitive modifiability through the visual mediation using games. We conclude therefore, that children exposed to the complexity experience of the games, shows increased cognitive skills such as spatial navigation, reasoning, memory and three dimensional perceptions.

3 EXPERIMENTAL SETUP

3.1 Development Process

The process of development of the video game with Semiotics was based on the life cycle of XP that performs iterative and incremental tasks (Beck, 2000), (Schneider, 2003). The research team carried out an incremental delivery of the product in each iteration.

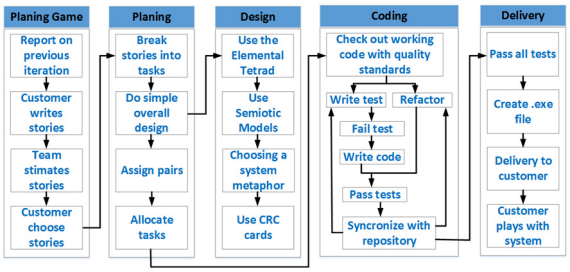


Figure 1: The Extreme Programming iterations game.

The experiment considered three iterations: (1) The design and development of the graphic user interfaces of the video game, for which we applied elementary Tetrad Game Design modified Model; (2) The construction of the inference engine, based in technical heuristics of Artificial Intelligence implemented with numerical methods that generate different levels of difficulty in the game; (3) The processing and storage of information, that keeps the users scores in files, considering the three levels of difficulty. In these iterations, liberated parts of the product were inspected and evaluated to increase the functionality and also to improve the quality compared to the previous versions of the game (Villacís, 2015), which has been implemented without using Semiotics. Fig. 1 shown the XP iterations game model modified based in the proposed model by Drake et al. (2006).

3.2 Design and Development of the GUI

The design and the development of the graphic user interface of the Tic-Tac-Toe video game were based on the Elementary Tetrad Game Design Model (Schell, 2014). For each one of the four sections of this model, we considered a series of elements related to programming computer games proposed by Walnum (2001), among them are: (1) Game design; (2) Graphic design; (3) Controls and interfaces; (4) Generation of sound; (5) Image handling; (6) Animation; (7) Algorithms; (8) Artificial Intelligence; (9) Game Testing. Additionally, it was necessary to include the Storyboard proposed by Páez (2013). Based on all of these elements, we propose the modified model illustrated in Fig. 2. The Mechanics section includes game algorithms and Artificial Intelligence algorithms. The Technology section includes graphic design, animation and image handling. The Aesthetics section includes control and interfaces, sound generation and game testing. The story section includes game design and storyboard.

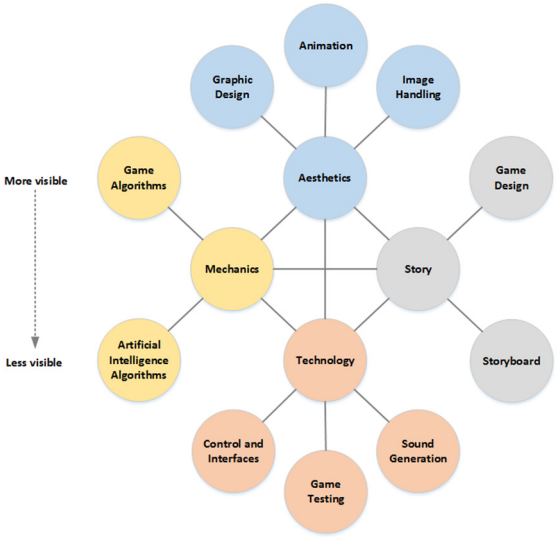


Figure 2: A proposal of the modified Elemental Tetrad Game Design Model for this study.

3.2.1 Game Design

The concept was based on real-world events related to the design and creation of boards for the game of Tic-Tac-Toe. This is a fun game that appeals to both children and adults because it is a game of strategy.

3.2.2 Graphic Design

The game screen of Tic-Tac-Toe has a form programmed in C# .NET, which is very similar to the dialogue frames of the Windows System. In this form is placed a series of objects that allowed to structure the game, such as: (1) Nine buttons for checkers, mathematically located in specific locations; (2) Four buttons to manage the game options (i.e. New, Choose players, Choose language and Exit); (3) Four group boxes or containers (Group Box); (4) One label for static text; (5) Four radio buttons to control the different levels of the game; (6) One picture box, which shows an animated icon; (7) A menu bar with File, Help and View options, and their respective submenus.

3.2.3 Control and Interfaces Implementation

The GUI was constructed based on components (COM+), and basic controls (ActiveX) that provided the Visual C# .NET programming language. The scores in the game were stored in XML flat files, and the data are displayed within the Data Grid View control. Fig. 3 illustrates the GUI of the game:

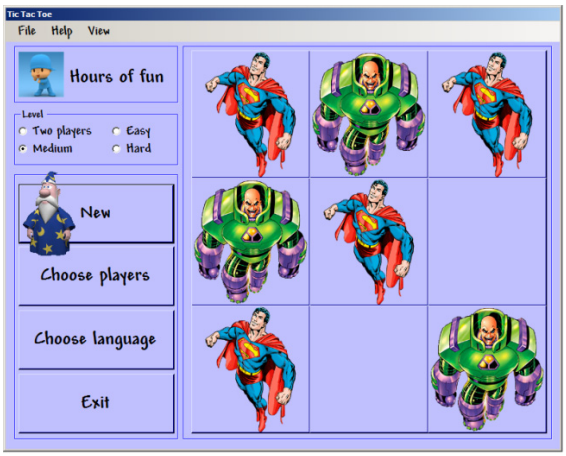


Figure 3: Graphical User Interface of the Tic-Tac-toe game using Semiotics.

3.2.4 Image Handling

We organized all the visual interfaces to handle images via ActiveX controls. For example, the Picture Box control allows loading animated gif files and the Button control is designed and constructed with geometric shapes which can load images in various formats. We have defined four categories in the game related to the nine buttons of the board, which are available for the user being: (1) *Super hero*: Selection between twelve super heroes and twelve villains; (2) *Princesses*: Selection between twelve princesses of fairy tales and twelve villains of those same fairy tales; (3) *Animals*: Selection between twenty four animals among which are both wild and domestic animals; (4) *Miscellany characters*: Selection between several well-known children's characters with their respective antagonistic characters. Finally no special background (i.e. Background Image property) was included, only colours, which is more attractive to children and generates much less distraction.

3.2.5 Sound Generation

Because the real world is a place with sound, the game needs to include sound so that it seems realistic. The Tic-Tac-Toe game is not the exception. Therefore, MIDI sounds type was implemented using Windows Media Player control.

3.2.6 Animation

A virtual assistant and speech recognition libraries for both: Spanish and English language were implemented using MS Agents by Microsoft.

3.2.7 Game Algorithms

One of the most important algorithms in the game is the algorithm that allows two users to play each other on the same computer or a user to play against the computer with AI. Depending on the level that the user chooses for which the function Play() has been implemented inside the form frmTicTacToe whose algorithm is presented at the end of this subsection.

The parameters of the managed objects explained above are: (1) *board*: whose value depends on the category chosen by the user; (2) *btnTicTacToe*: whose value depends on which the button has been clicked by the user; (3) *type*: whose value depends on the fictitious good or bad character chosen by the user; (4) *player1*: whose value depends on the character, animal or object chosen by the user based of the type and the board; (5) *player2*: this parameter represents the non-player character (NPC) selected by the user for the confrontation; (6) *grbTicTacToe*: this control represents the container that hold the nine buttons and when the game is over this control disables these buttons; (7) *dgvData*: this control shows the final results of the players saved on XML files; (8) *listButtons*: this parameter is used exclusively by the non-player character (NPC) in which case the button is selected depends on the heuristics techniques of Artificial Intelligence.

3.2.8 Artificial Intelligence Algorithms

Within the context of this study, the Artificial Intelligence algorithms focuses on providing capacity to computers to perform tasks that require human intelligence. This means, the ability of the computer to act or participate as an opponent in the game (Walnum, 2001). In the Tic-Tac-Toe game, the computer has the ability to play with the user, according to three different levels of difficulty: basic level, intermediate level, and advanced level. For the Artificial Intelligence model of the application we have used both weak and strong heuristics techniques. Here we use numeric method based on numeric series that is represented by linked lists and arrays. These kind of structures store different movements made by the same application that is the non-player character (NPC) controlled by the computer that plays with the user. The numeric method based on finite series is indicated in Table 1, where each finite series has been obtained based on a sum that represents a value accumulated in a certain row, column or diagonal of the Tic-Tac-Toe game. In Table 2, the initial state of the whole array is depicted (i.e., that is zero) and it corresponds to an empty space or a free cell. Some cases are described below:

Table 1: Numeric method based on finite series.

Rows	$\sum_{i=0}^{n=2} f_i = a$	$\sum_{i=3}^{n=5} f_i = b$	$\sum_{i=6}^{n=8} f_i = c$
Columns	$\sum_{i=0}^{n=6} c_i = d$ Step 3	$\sum_{i=1}^{n=7} c_i = e$ Step 3	$\sum_{i=2}^{n=8} c_i = f$ Step 3
Diagonals	$\sum_{i=0}^{n=8} d_i = g$ Step 4	$\sum_{i=2}^{n=6} d_i = h$ Step 2	
Diagonals (Trivial Case)	$\sum_{i=0}^{n=8} d_i = x$ Step 4	$\sum_{i=2}^{n=6} d_i = y$ Step 2	
Edges (Trivial Case)	$\sum_{i=2}^{n=5} t_i = p$ Step 3	$\sum_{i=1}^{n=3} t_i = q$ Step 2	
	$\sum_{i=5}^{n=7} t_i = r$ Step 2	$\sum_{i=3}^{n=7} t_i = s$ Step 4	

- **Case 1:** The non-player character (NPC) obstructs the user. In this case the following instruction should be considered: if ((a = 2) ∨ (b = 2) ∨ (c = 2) ∨ (d = 2) ∨ (e = 2) ∨ (f = 2) ∨ (g = 2) ∨ (h = 2)) then: if (v[k] = 0) then v[k]:= 3 → NPC obstructs the user;
- **Case 2:** The non-player character (NPC) wins. In this case the following instruction should be considered: if ((a = 6) ∨ (b = 6) ∨ (c = 6) ∨ (d = 6) ∨ (e = 6) ∨ (f = 6) ∨ (g = 6) ∨ (h = 6)) then: if (v[k] = 0) then v[k]:= 3 → NPC beats the user;
- **Case 3:** Obstruct in the diagonals. In this case the following instruction should be considered: if ((x = 5) ∨ (y = 5)) then: if (v[k] = 0) then v[k]:= 3 → NPC obstructs the user;
- **Case 4:** Obstruct in the corner squares. In this case the following instruction should be considered: if (((p = 2) ∨ (q = 2) ∨ (r = 2) ∨ (s = 2)) ∧ (v[k] = 0)), where k = 0, 2, 6, 8; then: if (v[k] = 0) then v[k]:= 3 → NPC obstructs the user in the corners close to the edges occupied by the user.

Table 2: Finite State Machine of the Game.

Object	Weight
User 1	1
Non-player character (NPC)	3
Blank space	0

3.2.9 Semiotic Model of the Game

Figure 4 shows the model of Aleferenko (Tokarev, 2014) which completes: K=S+O+D+C; where K is the Concept, S is the Sign, O it is the object, D it is the denotation and C it is the connotation. This model demonstrates the real state of the construction of the knowledge on the receiver's part since it is unable to jump the denotation and connotation process to arrive at the concept.

In the first instance, the sign is the word that corresponds to the game. In our case the sign begins the code process to obtain knowledge. In the second instance, the object referent corresponds to the code of the game. In the third instance, the receiver passes to the connotative process where it identifies the signs as symbols. Therefore it is not an arbitrary process of significance. The connotation process is in the fourth instance, where the receivers are the children that play the game, in an appropriate atmosphere. Also, according to their preferences, the sign that doesn't has the character of arbitrary in this case because they become symbols. For instance, in the case of the Superman – sign (the symbol of a super hero), which would correspond to K that is the concept of the sign S, and in similar form is Lex Luthor – sign (the antagonistic or villainous symbol with respect to superman). Finally, the result is the concept or knowledge decoded due to the process of connotation of the sign.

3.2.10 Principle of Arbitrariness of the Sign

According to Holdcroft (1991), the principle of arbitrariness of the sign consists in the bond among the meaning with the significance of arbitrary. This, since the sign is equivalent to the association of a significant with a meaning. For this reason the game of the Tic-Tac-Toe is decoded like it was mentioned previously, being a matrix with two elements: a 'cero' and an 'x' letter, generally constructed in paper or with wood. The concept or significance of Tic-tac-Toe is not bound for any relationship with the sequence of sounds "t-i-c-t-a-c-t-o-e" that serves by itself significance to the word. It could be represented by any other sequence of sounds, for example the "Tic-Tac-Toe" game in English corresponds to "Tres en Raya" in Spanish and in Russian, it corresponds to "крестики-нолики", where it doesn't only change the sound but even the system of signs that doesn't correspond to the Latin alphabet, and the system corresponds to the Cyrillic one instead.

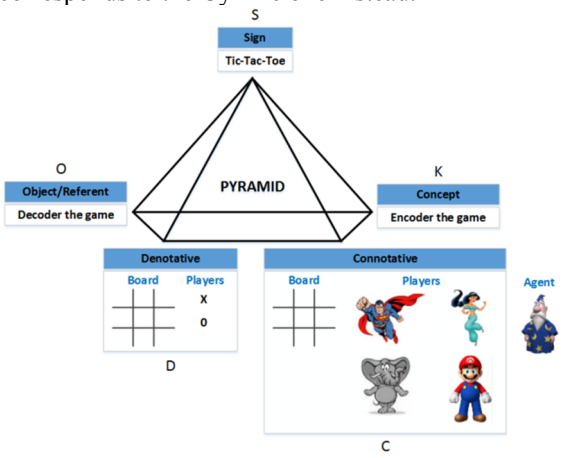


Figure 4: The semiotic model Aleferenko for the game.

3.2.11 Testing the Game

We applied the game to a public school, forty seven to eleven years old children were randomly chosen to play the game during 30 minutes. A group of children were exposed to the traditional Tic-Tac-Toe game (i.e. without Semiotics) versus the optimized Tic-Tac-Toe game (with Semiotics). After, the children tested the game, we proceeded to perform statistical processing of the scores provided by the game. As noted in this study, it has been optimized as the Tic-Tac-Toe game, incorporating the semiotic model of Aleferenko. This research has involved superheroes, princesses, other animals and world comics.

4 EVALUATION RESULTS AND DISCUSSION

4.1 Results of the Evaluation

There were differences of comic figures preferences. Fig. 5 illustrates that there is a superhero that has the popularity of 100% (i.e. Hulk), followed by another one with 81.8% (i.e. Spiderman). The other two with 36.4% (i.e. Superman), and Wonder Woman with 27.3%. Batman and Green Arrow with 18.2% and 9.1%, respectively. This means that two superheroes were known to more than half of the children with more than 50% of the total popularity, while 14 other superheroes yielded amounts between 18.2% and 36.4% on average, and just six below 9.1%. This leads to the conclusion that 16 boys were interested in recreational games with superheroes.

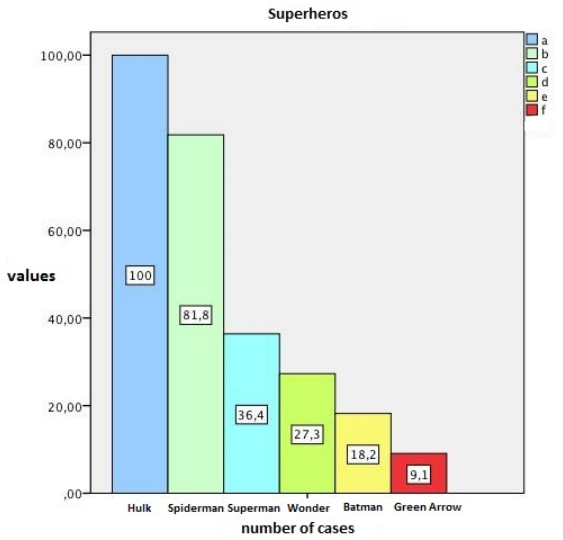


Figure 5: Bar chart of the most popularly superheroes selected in the game.

In the case of girls the results were: Two princesses reached total popularity of 100% (i.e. Anna and Elsa), while five reached popularity amounts between 16.7% and 41.7% (i.e. Rapunzel, Ariel, Goethel, Jasmine, and, Cinderella). These amounts are all above average. However 17 are almost unknown to the children as they reached popularity values below 8.3% (Snow White, Pocahontas, Queen Grim Hilde, Shan Yu, Hans, Bella, Mulan, Tiana, Merida, Aurora, Lady Tremaine, Ursula, Jafar, John Ratcliffe, Dr Facilier, Queen Elinor, and Maleficent).

The results accomplished by children in the fifth grade of elementary school at the intermediate level

shows that the modified Tic-Tac-Toe game (with semiotics) conducted the successful challenge by the children., but gender difference were detected. Fig. 6 illustrates that in the normal game did not crystallize any winners, but there were three losses and three ties (i.e. equal finish). The girls in turn drew all. On the other hand, Fig. 7 demonstrates two boys won, having just three losses and one draw. Girls continue getting the same draws. However, in the case of girls no improvement could be reached. This analysis was performed in each grade and with different levels (i.e. basic, intermediate and advanced) demonstrating better results with the game optimized with Semiotics, contrasting the research hypothesis convincingly.

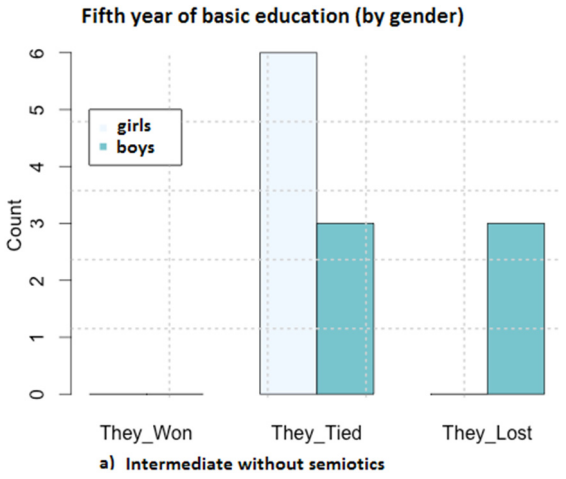


Figure 6: Results obtained using Tic-Tac-Toe game without semiotics.

Finally, the illustration in Fig. 8 has been obtained by the means of the method of Natural and Hyman. In this figure it is demonstrated, that the frequency curve of the normal variable change fairly its tendency compared to the curve generated by the semiotic variable. This leads to the conclusion, of the existence of a greater homogeneity of the data of the semiotics variable compared with normal variable. Thus, we are able to suggest, that it is more difficult to resolve recreational games with normal conditions when a semiotic modelling applies. Therefore, we deduce that semiotics grows proportionally affecting more positively the learning in an objective and scientific way. Similarly, based on the study conducted at different grades (second to sixth grade) in an elementary school we obtained a similar behaviour.

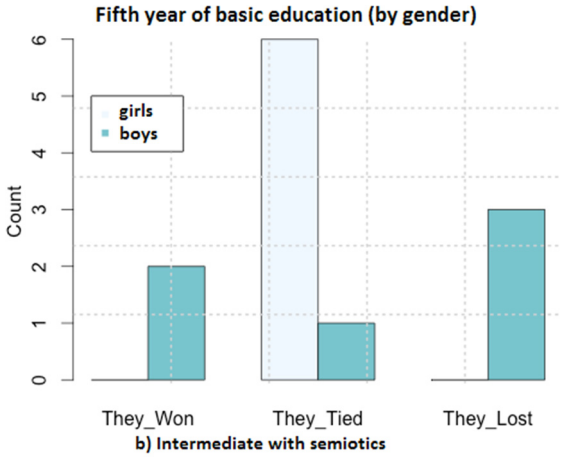


Figure 7: Results obtained using Tic-Tac-Toe game with semiotics.

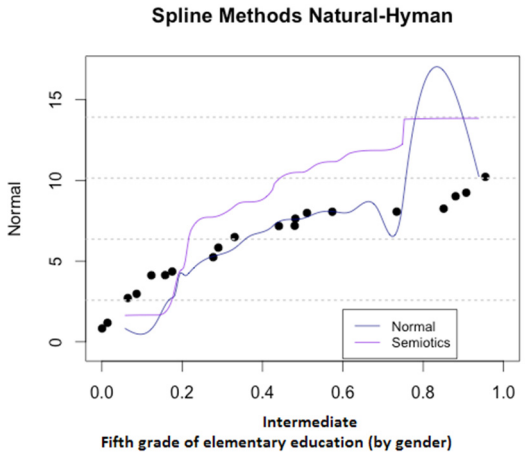


Figure 8: Results obtained adjusting an algebraic polynomial from the absolute cumulative frequencies.

4.2 Discussion

It is considered that the Aleferenko model integrates the foundations of Semiotics. Another important fact that carried this model to the desktop platform game is its friendly appearance. Therefore, it has been empowering for the comprehensive knowledge of children between 7 and 11 years old and can be used as part of the teaching-learning process of schoolchildren, because this game is a strategy educational game that is using to stimulate logical and spatial reasoning. In this study, that was achieved due to the implication of connotative analysis in deep through Artificial Intelligence and Software Engineering, which causes the receiver (user) uses tactics and strategies to beat the computer. Thus, a dynamic environment was created in the game for the

receivers who can have their own signs according to the age. They lose their arbitrariness and the symbols become the identifiers with which the children are related and can create relevance and internalization of the knowledge, and above all to create interactivity. Another important aspect when evaluating the game includes the use of signs and symbols and their diagnostic meaning for the children. Therefore, this creates a natural, interesting, efficient and motivating learning.

5 CONCLUSIONS

This research focused on how to optimize an educational video game named Tic-Tac-Toe by means of semiotics analysis, in order to stimulate logical and spatial reasoning of children. The main issue in our study has been to design a mathematical model, which is implemented with AI algorithms and a graphical user interface including Semiotics, applied to an incremental methodology with the aim of producing an enjoyable and interactive environment. To carry out this study, we have used the Extreme Programing (XP) agile methodology for the codification and testing of the incremental products in each iteration (sprint), combined with the modified Elemental Tetrad Game Design Model for defining the performance of the game's models, in order to ensure the quality of the software used. To validate our results, the proof of concept and testing has been performed with a representative sample at an elementary school, focused on children with ages ranging from seven to eleven years old. The results imply that educational video games with Semiotics stimulate cognitive development of children.

REFERENCES

Gee, J. P., 2008. Learning in semiotic domains. *Literacies, global and local*, 137-149.

Halliday, M. A. K., 1978. Language as social semiotic: The social interpretation of language and meaning. Maryland. University Park Press.

Myers, D., 1991. Computer game semiotics. *Play & Culture, 4(4)*, 334-345.

Thorne, S. L., Fischer, I., & Lu, X., 2012. The semiotic ecology and linguistic complexity of an online game world. *ReCALL, 24(03),* 279-301.

Kendall, N., 2015. What is a 21st Century Brand? New Thinking from the *Next Generation of Agency Leaders*. Kogan Page Publishers.

Huber, W. H., 2013. *The Foundations of Videogame* Authorship.

Ruiz, J. G., et al., 2014. Evaluating the Communicability of a Video Game Prototype: A Simple and Low-Cost Method. *In Proc. of the 5th Mexican Conference on Human-Computer Interaction* (p. 30). ACM.

Baceviciute, S., Bruni, L., 2014. Experience Cognitive Experiences, ICoN2014 - *1st International Congress of Numanities.* The role of Humanities in Contemporary Society: Semiotics, Culture, and Tech.

Villacís, C., Fuertes, W., Bustamante, A., Almachi, D., Procel, C., Fuertes, S., & Toulkeridis, T. 2014. Multi-player Educational Video Game over Cloud to Stimulate Logical Reasoning of Children. In Proceedings of the *IEEE/ACM 18th International Symposium on Distributed Simulation and Real Time Applications (pp. 129-137).* IEEE Computer Society.

Gibson, J., 2014. Introduction to Game Design, Prototyping, and Development: *From Concept to Playable Game-with Unity and C#.* Pearson Educ.

Beck, K., 2000. Extreme programming explained: embrace change. Addison-Wesley Professional.

Schneider, J. G., & Johnston, L., 2003. Extreme Programming at universities: an educational perspective. *25th International conference on software engineering (pp. 594-599).* IEEE Computer Society.

Tokarev, G.B., 2014. Introducción a la Semiótica, Flinta Nauka, Moscú, pág. 62.

Villacís, C. J., Fuertes, W. M., Bustamante, C. A., Zambrano, M. E., Torres, E. P., Aules, H. M., & Basurto, M. O. (2014). Optimización del juego tres en raya con niveles de dificultad utilizando heurísticas de inteligencia artificial. *AtoZ: Novas Práticas em Informação e Conhecimento, 3(2),* 95-106.

Wisse, P., 2002. Semiosis & Sign Exchange: *design for a subjective situationism,* including conceptual grounds of business information modeling.

Vygotsky, L. S., 1967. Play and its role in the mental development of the child. *Soviet psychol, 5(3),* 6-18.

Brown, A. L., & Ferrara, R. A., 1999. Diagnosing zones of proximal development. Vygotsky: Critical assessments: *The zone of proximal dev., 3,* 225-256.

Feuerstein, R., Klein, P. S., & Tannenbaum, A. J. (Eds.), 1991. Mediated learning experience (MLE): Theoretical, psychosocial and learning implications.

Lipman, M., & Sharp, A. M., 2002. *La filosofía en el aula (Vol. 4).* Ediciones de la Torre.

Drake, P., & Kerr, N., 2006. Developing a computer strategy game in an undergraduate course in software development using extreme programming. *Journal of Computing Sciences in Colleges, 22(2),* 39-45.

Schell, J., 2014. The Art of Game Design: *A book of lenses.* CRC Press.

Walnum, C., 2001. Sams Teach Yourself *Game Programming With Visual Basic in 21 Days*. Sams.

Holdcroft, D., 1991. Saussure: signs, system and arbitrariness. Cambridge University Press.

Towards a Reference Architecture for Advanced Planning Systems

Melina Vidoni and Aldo Vecchietti
Ingar Instituto de Desarrollo y Diseño, CONICET-UTN, Avellaneda 3657, Santa Fe, Argentina
{melinavidoni, aldovec}@santafe-conicet.gov.ar

Keywords: 4+1 View Model, Advanced Planning Systems (APS), Functional Requirements, Reference Architecture.

Abstract: Advanced Planning Systems (APS) are important for production companies that seek the optimization of its operations. However there are gaps between the companies' needs and its implementation in the Enterprise Systems, such as the lack of a commonly accepted definition, the short insight on its software architecture, and the absence of Software Engineering (SE) approaches to this type of system. Consequently, it is important to study APSs from a SE point of view. The motivation of this work is to present a Reference Architecture for APS, providing a standard-based characterization and a framework to simplify the design, development and implementation of APS. Therefore, two views are presented, which are based on the "4+1" View Model endorsed by the international standard ISO/IEC 42010:2011; those Views are represented using UML diagrams and they are described including variation points for a number of possible situations.

1 INTRODUCTION

Advanced Planning Systems (APS) are part of many organizations and are linked to the Enterprise Systems (ES) aiming to optimize raw materials, inventory, production plans, etc., to improve the economy of the company (Stadtler, 2005).

Some high-end ERP (Enterprise Resource Planning) offer extra modules to perform APS functionalities customized and adapted to each business. Examples are SAP APO (Advanced Planning and Optimization) (Stadtler, et al., 2012), and Oracle ASCP (Advanced Supply Chain Planning) (Oracle, 2015). However, on small and medium enterprises the most common implementation approach is an *ad-hoc* development, performed inside the house or outsourced.

Thus, there is interest on a better understanding of several issues related to the development of APS (Zoryk-Schalla, et al., 2004), such as the lack of standardization in associated concepts (Kallestrup, et al., 2014; Aslan, et al., 2012; Hvolby & Steger-Jensen, 2010), and the lack of SE approaches (Henning, 2009; Framinan & Ruiz, 2010).

Recently, Vidoni and Vecchietti (2015) proposed an APS characterization, by applying a SE approach, and elicited Functional Requirements (FR) and Quality Attributes (QA) from the academic literature which is used to elaborate a Reference Model for the Software Architecture (SwA) of an APS.

Still, a Reference Model is a starting point and needs to be upgraded into a Reference Architecture (RA) (Northrop, 2003). The latter are abstractions of specific SwA for a given domain, and are used as standardized frames or tools (Angelov, et al., 2012).

There are many researches about RA. Norta et al. (2014) introduced one for Business-to-Business systems, for research and industrial applications. Pääkkönen et al. (2015) proposed a RA for big data systems, based on the analysis of architectures previously implemented. Behere, et al. (2013) announced a RA for cooperative driving on modern vehicles with a minimally invasive model. Finally, Nguyen et al. (2011) developed a RA based on the "4+1" View Model, to define agent-based systems.

This work proposes the first two views towards a RA for the APS, based on the "4+1" View Model (Kruchten, 1995), recommended by the international standard ISO/IEC 42010:2011 (2011). The work is based on the FR and the Reference Model proposed in a previous work (Vidoni & Vecchietti, 2015). A comparison of the FR with leading commercial suites is summarized, to prove its applicability.

This paper is structured as follows. Section 2 introduces concepts and definitions, and Section 3 presents the FR elicited for APS, comparing them to features of leading commercial suites. Section 4 introduces the RA concepts, standards and design decisions, describing two views. Finally, Section 5 presents conclusions and related future works.

Vidoni, M. and Vecchietti, A.
Towards a Reference Architecture for Advanced Planning Systems.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 433-440
ISBN: 978-989-758-187-8

2 CONCEPTS AND DEFINITIONS

A definition for APS is the one given by Stadtler (2015), which states: *"Although an Advanced Planning System (APS) is separated into several modules, effective information flows between these modules should make it a coherent software suite. Customizing these modules according to the specific needs of a supply chain requires specific skills, e. g. in systems and data modeling, data processing and solution methods. APS do not substitute, but supplement existing ERP*".

This paper will also use the concept of *factory planning* (which includes several types of planning mostly at short-term) and *supply chain planning* (represents factory planning problems beyond the company limits, at mid and long term time horizons) introduced by (Fleischmann & Koberstein, 2015).

There are also other definitions considered:

- *Enterprise Systems* (ES), includes ERPs, transactional systems and other information systems that manages data in an organization (Davenport & Brooks, 2004).
- *Solving Approach* (SA), an umbrella term that refers to the advanced methods and technics used to solve advanced planning problems. Includes operations research, genetic algorithms, game theory, and others.
- *Optimization Point* (OP) is a specific planning problem solved through an APS.
- *Model* is a specific solution for an individual factory planning problem, using any SA.
- *Objective* is what the model seeks to optimize.

3 FUNCTIONAL REQUIREMENTS

Based on Software Engineering Body of Knowledge (BKCASE Editorial Board, 2014), the Functional Requirements of a system *"[…] describe qualitatively the system functions or tasks to be performed in operation; FR defines what the system must be able to do or perform"*.

Since this is a high abstraction level definition, there are no explicit stakeholders, and the hardware to be used in the architecture is undefined. Therefore, requirements were extracted from the academic literature related to APS, where they are usually presented as general statements and ideas.

Vidoni and Vecchietti (2015) introduced a list of generic FR elicited from the academic literature, and based on a number of international SE standards. These FR are descripted on Table 1, where each row represents a new requirement, with an ID code (first column at the left) later used as reference.

However, these FR are generic, suitable for a wide definition that can work as a frame (Angelov, et al., 2008). Therefore, not all of them must be met by each application of APS, because those specific implementations are sub-sets, carefully selected for those situations; new requirements can also be added, because a software intensive system never ceases to evolve and change.

Since leading proprietary suits are developed through several iterations and continuously refined, they implement many of the features characteristics of APSs, contributing to the "best practices" idea of leading ES. Therefore, by comparing the proposed FR, it can be seen that there is a high level of agreement, which supports their applicability, and provides the fundaments to develop the RA.

The selected commercial applications are leaders in the ERP market: APO (Advanced Planning and Optimization) for SAP ERP (Stadtler, et al., 2012), and ASCP (Advanced Supply Chain Planning) for Oracle e-Business (Oracle, 2015). Both of them are available as separated modules of their ES solutions, and have available online documentation.

Evaluating SAP APO reveals a high match of the application's features to the FR. APO works with two planning levels: Supply Network Planning (SNP) is midterm/long term planning, while Production Planning/Detailed Scheduling is short term, similar to the factory and supply chain planning concepts introduced before.

SNP module optimizes several OP, allowing selecting the SA and model versioning, changes and adaptation. Users can optimize while working on the system in parallel; results are in friendly manner and include historical data. As input data APO uses demand planning, sales orders, and even ETO, transferred from SAP ERP via the Core Interface; the approved output data is also stored on the ERP, while the other is kept on the APO's own database. APO checks consistency and bottlenecks, and evaluates rescheduling (Stadtler, et al., 2012).

By studying Oracle e-Business ASCP features, it is clear that this suite reinforces them, with a different approach than SAP APO, considering Usability as one of the main QA of the system. ASCP allows several OP and models, with management functions including many settings

Each model has available objectives, and parameters management. Each planner (user) can configure the interface, while the Planner Workbench offers scenario comparison. ASCP uses

Table 1: Summary of Functional Requirements (Related to the System) for an APS (Vidoni and Vecchietti, 2015).

Code	Requirement	Requirement Description (Related to the System)
A	Optimization Point Management	The system must have at least one Optimization Point, and there is no limit to how many may optimize. The user must be able to select which Point to work with at any given time. Each OP (which represents a planning problem) has at least one model that solves it.
B	Models Management	The system must allow the user to easily select the model to be used on each OP. If there is more than one model, the APS should have a default, if nothing else was selected.
C	Objectives Management	The system allows the user to select the objective to use with each model. Each model must have a default objective that will be used in case no other one was manually selected.
D	Parameters Settings	The system must offer a graphical way for the user to customize the parameters (changing values, ranges and increments). In case no value was changed, it must use the defaults.
E	Scenario Generation	After the used input of the parameters, the APS must automatically generate each scenario, showing progress to the user and allowing them to continue with other tasks.
F	Scenario Storage	The scenarios results must be automatically stored (in either success or failure/infeasibility situations) on the APS database, to be later revised and studied by the human planner. Results are only impacted on the ES once the user approves them.
G	Scenario Comparison	The system must offer a Graphical Interface (GUI) to compare scenario results and allow the planner user to modify them. For successful cases the comparison should show charts, graphics, statistics of resolution times, and so on. For unfeasible results, the showcased information must help the planner to understand why the model turned unfeasible.
H	Input Data	The APS must automatically read the input data for each model from the ES.
I	Consistency Check	There must be an evaluation of the data entered on the system before running each model. This checks the existence of needed resources, including availability of raw materials, comparing Bills-of-Materials to current stock, machine states, and so on. If the check fails it means that the solution was possibly unfeasible, and it must be informed to the planner.
J	Output Data	The system translates the results of the selected scenario to a format understood by the ES, and stores it on it. This is only done when approved by the user.
K	Log-in Function	The APS restricts access to authorized-only personnel.
M	Open/Saving Results	The system should be able to open and show previous results with the same charts, graphics and displays used before, during the comparison.
N	Algorithm Integration	An authorized user must be able to perform CRUD (Create, Read, Update, Delete) actions for the components (models, objectives, parameters values) of each optimization point.
O	Bottleneck Detection	The system should check bottlenecks and under-loaded resources, with the aim of avoiding proposing a planning that is not optimal regarding the use of resources. If any issue is detected, it must be penalized and/or informed to the user, awaiting their input.
Q	Rescheduling Checking	After a deviation from the plans, the system should show whether the current jobs have to be rescheduled. This should be decided by the human planner, or an automatized option.

any input data synchronized from any ES (forecasts through an external module, sales orders and ETO data), and allows deciding where to store the output data. It also provides bottleneck detection (Oracle, 2015).

4 REFERENCE ARCHITECTURE

In SwA, Reference Model (RM) is a division of functionalities with data flow between pieces, working as a standard decomposition of a known problem. Then, a RA is a RM mapped onto software elements that cooperatively implement the FR, and the data flows between them (Northrop, 2003).

An RA is presented with standardized diagrams that describe it through a number of viewpoints, fulfilling the needs of different stakeholders; these abstract the detail of implementation, detailing relations between components (Yonghua Zhou, et al., 2004). However, their generic nature leads to less defined contexts, increasing the design complexity; consequently, it is a non-trivial matter, surrounded by ambiguity (Angelov, et al., 2012).

The ISO/IEC 42010 (2011) standard enforces the application of viewpoints to clarify different approaches in a system description through a RA. In particular, Annex B recommends the "4+1" View Model (Kruchten, 1995), selected for this work. It consists of four views (Logical, Development, Process, Physical) and the "+1" represents Scenarios, based on the FR. Since the model allows the use of any standardized diagram, UML 2.x (Object Management Group, 2013) is selected due to its widespread use. Since there is no direct match between diagrams and views, this work will follow the associations presented in other papers (Nguyen,

et al., 2011; Kontio, 2008).

Because this is a work in progress, only two views are presented, with the documentation pattern by Bachmann, et al. (2003): introduction with UML diagram, a description of elements and relations, a variability guide and an architectural background. The latter adds variation points to allow variability in the RA, to accomplish modifications in pre-planned ways, adding changes during development in specific study cases (Clements, et al., 2010).

4.1 Logical View

This view supports the FR, showing what the system should provide as services to its users (Kruchten, 1995); the elements are "key abstractions" manifested as objects, components or packages (Northrop, 2003). This is the first view developed, and is translated from the FR and RM of a previous work (Vidoni & Vecchietti, 2015).

4.1.1 Primary View

Fig. 1 presents the Logical View for the RA, using a Model Diagram. This is an auxiliary UML structure diagram that shows an abstraction or specific view of a system, describing its architectural, logical or behavioural aspects (Object Management Group, 2013). Model Diagrams uses Package Diagram syntax, and represents logical aspects of the layered APS system, and the actors that relate with it.

4.1.2 Architecture Background

Table 2 shows a match between the FR and the blocks of the RM (Vidoni & Vecchietti, 2015), to

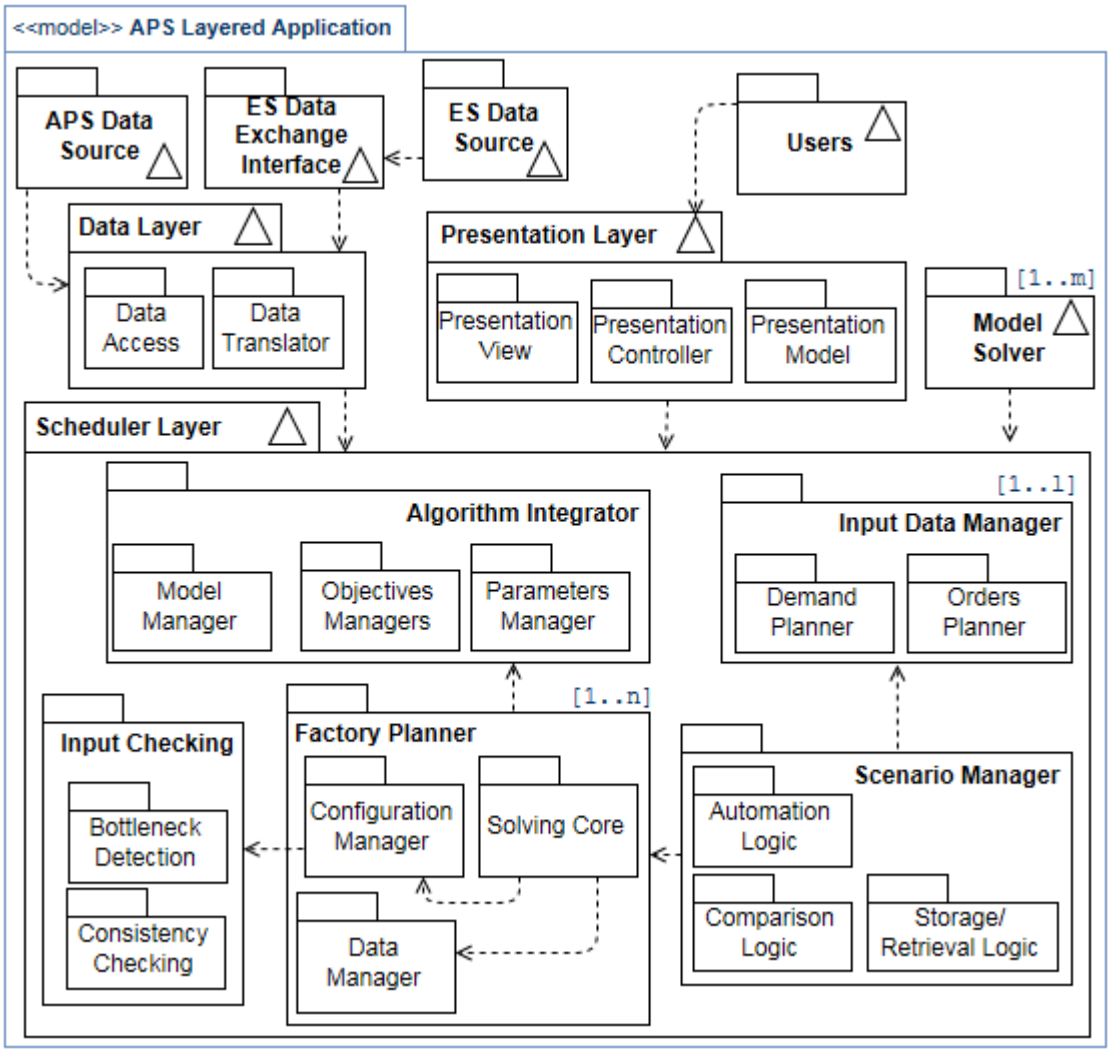


Figure 1: Model Diagram for the Logic View of the APS-RA.

the packages of the Logic View. The FR are grouped considering their relations, their need to interoperate, or if they are part of a bigger workflow. Both databases (APS's and ES's) are actors, along with the Model System, which represents a variation point. The Logic View has new packages, added in order to clarify the sorting of the FR, providing more helpful blocks to the view stakeholders.

For example, Scenario Manager represents many FR (by code: F, G, M, and the automation of the solving in 'Scenario Generation'). It represents a breakdown of the RM block 'Scenario Manager', and is essential for an APS. Also, the RM Factory Planning block (along with the FR coded as A, E and N) translates to the new Factory Planner, which is composed of: Solving Core, Data Manager and Configuration Manager.

4.1.3 Element Catalogue

There are five actors: Users (human planers and decision makers that use the APS), APS Data Source (the APS database, mentioned on some FR), ES Data Source (ES-DS, represents the ES database, and is mentioned on a many FR), ES Data Exchange Interface (ES-DEI, represents an optional interface provided by the ES for database access) and Model Solver (MS, represents external systems that solves models of any SA, providing raw results).

A, example of ES-DEI is the Core Interface used by SAP APO to exchange information with SAP ERP (Stadtler, et al., 2012). Also, the cardinality $(1..m)$ in indicates that there can be multiple instances of this actor, and the amount is not directly related to how many OP exists within the APS.

APS is composed of three main layers (Presentation, Scheduler, Data) representing a logical distribution of the code with a theoretical base (Lothka, 2005). The layers can be arranged in tiers, depending on implementation decisions -such as infrastructure, users, geographic distribution, etc.- (Microsoft Patterns & Practices Team, 2009) that are outside the abstraction level for a RA.

The *Presentation Layer* includes Graphical User Interface, and considers implementation specifics, by showing an inner Model-View-Controller pattern. *Data Layer* groups the database management logic and translation, and relates to data sources actors, either directly or through the ES-DEI.

The third layer is the APS core: *Scheduler Layer*. The content is grouped on four packages, which covers the main FR: Input Data Manager (main logic to obtain input data: forecast through Demand Planner, and MTO/ETO through Order Planner), Input Checking (contains evaluation logic, including the FR I and O), Factory Planner (core logic for each OP to be solved; includes data translation, outsourcing to MS, and point configuration), Algorithm Integration (create/read/update/delete functions for models and components), and Scenario Manager (automation of scenario generation, grouping requirements E, F, G).

4.1.4 Variability Guide

The actor ES-DEI is a variation point, because it only exists if the ES offers an interface, or if it was developed in the organization; the most complex

Table 2: Matching between elicited FR, original RM blocks, and current packages from the Logic View of the RA.

Functional Requirements	Reference Model Blocks	Logic View Packages
S: Database Use	APS Database Control	APS Data Source
K: Output Data L: Information Exchange	ES Database Control	Package: Data Access. External Systems: ES-DS and ES-DEI
B: Models Management, C: Objective Management D: Parameters Setting N: Algorithm Integration		Algorithm Integrator: ▪ Model Manager ▪ Objective Manager ▪ Parameters Manager
H: Input Data L: Information Exchange	Demand Planning	Input Data Manager: ▪ Demand Planner, Orders Planner
I: Consistency Check O: Bottleneck Detection	Consistency Checking	Input Checking: ▪ Consistency and Bottleneck Checking
E: Scenario Generation F: Scenario Storage G: Scenario Comparison M: Open/Saving Results	Scenario Manager	Scenario Manager: ▪ Storage/Retrieval Logic, Comparison Logic and Automation Logic
A: Optimization Points Management E: Scenario Generation N: Algorithm Integration	Factory Planning	Factory Planner ▪ Configuration/Data Manager, Data Manager and Solving Core

relation is included. Also, there can be multiple MS actors when several SAs are used, or when models need different solvers. Since specifics of the connection and translation between MS and APS are outside the boundaries of a RA, only an umbrella actor is depicted on the view.

'Input Data Manager' represents another variation. In the case of an MTS model, it uses 'Demand Planner', while a MTO/ETO connects to 'Orders Planner'. How many instances or implementations of this module are needed, depends on the OP and their models. Also, 'Demand Planner' may manage more than one type of forecasts, and 'Orders Planner' may read multiple types of orders.

'Demand Planner' can also be an external system (like in Oracle ASCP case) that must interoperate with the APS. The APS-RA considers it as an internal package, like it is on SAP APO.

4.2 Development View

This view focuses on the actual modules organization, at the software environment, packaged in sub-systems and components; it helps to allocate FR and manage the project development (Kruchten, 1995). It shows the organization of modules, libraries, subsystems, and development units, mapping software to environment (Northrop, 2003).

4.2.1 Primary Presentation

Using UML, a Component Diagrams (Object Management Group, 2013) represents the view, which can denote either logical (e.g. business or process modules) or physical elements (e.g. COM+ or .NET elements, etc.) (Fakhroutdinov, 2014). Which type of component is used depends on the required level of abstraction of the diagram.

The Component Diagram of Fig. 2 represents logical components, which may have different levels abstraction. The external actors that interoperate with the APS have been included as systems.

4.2.2 Element Catalogue

There are three systems that need to interoperate: MS and ES represent 'actors' of the Logical View, and they are specified in order to show how they relate. However, since these actors may vary for each specific SwA, they are also variation points.

The third system represents the APS itself. Both Presentation and Data Layers are mapped to subsystems, while the packages from the Scheduler layer are now subsystems, in order to increase readability, by avoiding adding more subsystems.

The connections to the ES and MS are made usually using a TCP/IP protocol, regardless if it is internet or intranet. While the connection is in the diagram, its implementation may vary on each case.

It is important to note the match between both views, because it displays their interrelation and shows the representation of logic components. Physical components are outside the scope of a RA (Behere, et al., 2013), and are not presented.

Developers are the stakeholders for the view, and thus, components are modular parts with encapsulated content (Fakhroutdinov, 2014), that are refined and modelled through the development life cycle. Since a component may be manifested by many artefacts, the current level of abstraction is enough for the RA, leaving enough room to add particular considerations for each concrete case.

4.2.3 Variability Guide

Since the 'Model Solver' represents the actor MS of the Logic View, it has the same condition than before, and more than one may exist; then, the connection/translation with the APS may vary.

The internal components of ES are detailed, because the existence of the ES-DEI component depends on the organization. This variation point can change the interoperation and communication between with the APS, and must considered.

The subsystem that represents the User Interface has a higher level of abstraction than in the Logic View, because at the component level, the inner components vary with some implementation decisions (such as programming language, and type of application -web, standalone, etc.).

Also, following previous decisions, 'Demand Planner' is showcased as an inner component of the 'Input Data Manager' subsystem. Still, it can be an external system, as it is on the Oracle ASCP case.

5 CONCLUSIONS

This paper presents work in progress towards a Reference Architecture for an APS (APS-RA), based on Functional Requirements previously elicited through a study of the literature.

The FR are compared to the main features of commercial leading suites (SAP APO and Oracle e-Business ASCP), to validate the proposed requirements, obtaining a good match between them. The paper introduces the first two views of the APS-RA, based on the "4+1" View Model, suggested by

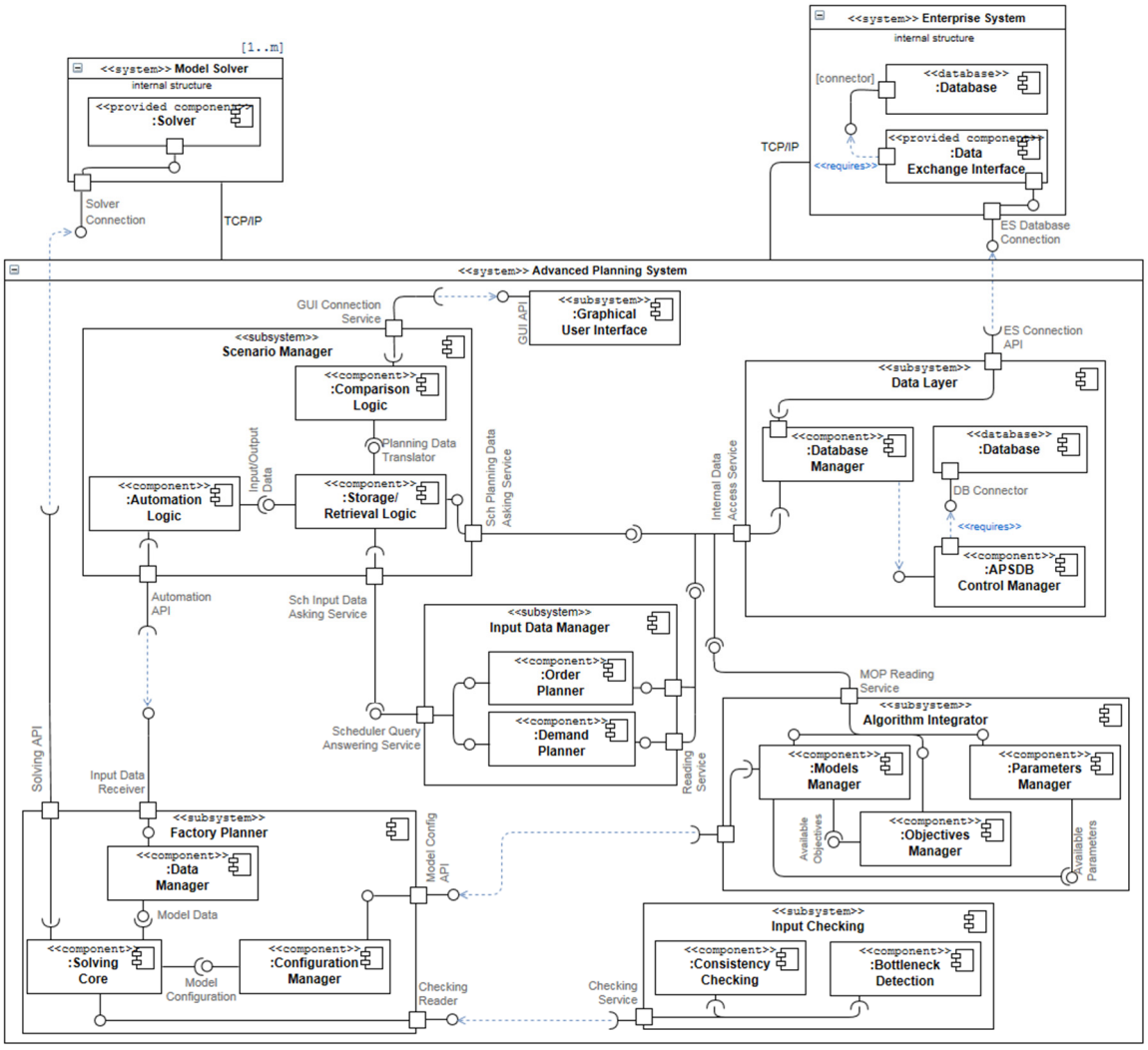


Figure 2: Component Diagram for the Development View of the APS Reference Architecture.

the standard ISO/IEC 42010:2011 for SwA, and is represented using UML 2.x diagrams. Only two views are introduced due to space limitations.

This work offers the beginning of a framework to support the implementation, helping to define and clarify the functionality of each component. It adheres to standardized SE methods, without adding load to the development process. This increases the quality of the development, providing the essential base for a clean design with intrinsic relations between FR, QA and the RA. It allows the project team to efficiently and effectively asses the quality and extensiveness of existing systems, guiding the modification and adaptation of existing systems to new developments.

Several lines for future works exist, besides completing the remaining views of the RA. The first of them is using the Quality Attributes that were previously elicited along with the FR to generate QA Scenarios and supplement them with metrics and indicators based on the international standard series ISO/IEC 2500n "Quality Management Series". With these, there is a third future work: evaluate the commitment of the APS-RA with those QA, by applying a Software Evaluation method, such as ATAM (Architecture Trade-off Analysis Method). A Final future work is to create a specific implementation of a study case, applying real-case data, and using the elements generated throughout this works (FR, QA, and the APS-RA).

REFERENCES

Angelov, S., Grefen, P. & Greefhorst, D., 2012. A

framework for analysis and design of software reference architectures. *Information and Software Technology,* 54(4), pp. 417-431.

Angelov, S., Trienekens, J. J. M. & Grefen, P., 2008. *Towards a Method for the Evaluation of Reference Architectures: Experiences from a Case.* Paphos, Cyprus, Springer Berlin Heidelberg, pp. 225-240.

Aslan, B., Stevenson, M. & Hendry, L., 2012. Enterprise Resource Planning systems: An assessment of applicability to Make-To-Order companies. *Computers in Industry,* 63(7), pp. 692-705.

Bachmann, F. et al., 2003. Chapter 9. Documenting Software Architectures. In: *Software Architecture in Practice.* 2nd ed. Boston, MA: Addison-Wesley.

Behere, S., Törngren, M. & Chen, D.-J., 2013. A reference architecture for cooperative driving. *Journal of Systems Architecture,* 59(10), pp. 1095-1112.

BKCASE Editorial Board, 2014. *The Guide to the Systems Engineering Body of Knowledge (SEBoK).* 1.3 ed. Hoboken, NJ: The Trustees of the Stevens Institute of Technology.

Clements, P. et al., 2010. Chapter 9. Beyond the Basics.. In: *Documenting Software Architectures. Views and Beyond.*. Second Edition ed. s.l.:Addison-Wesley, pp. 217-260.

Davenport, T. & Brooks, J., 2004. Enterprise Systems and the Supply Chain. *Journal of Enterprise Information Management,* 17(1), pp. 8 - 19.

Fakhroutdinov, K., 2014. *UML 2.x Diagrams.* [Online] [Accessed 2015] Available at: http://www.uml-diagrams.org/component-diagrams.html.

Fleischmann, B. & Koberstein, A., 2015. Chapter 6. Strategic Network Design. In: H. Stadtler, C. Kilger & H. Meyr, eds. *Supply Chain Management and Advanced Planning.* Germany: Springer Berlin Heidelberg, pp. 107-123.

Framinan, J. & Ruiz, R., 2010. Architecture of manufacturing scheduling systems: Literature review and an integrated proposal. *European Journal of Operational Research,* 205(2), pp. 237-246.

Henning, G., 2009. Production Scheduling in the Process Industries: Current Trends, Emerging Challenges and Opportunities. *Computer Aided Chemical Engineering,* Volume 27, pp. 23-28.

Hvolby, H.-H. & Steger-Jensen, K., 2010. Technical and industrial issues of Advanced Planning and Scheduling (APS) systems. *Computers in Industry,* 61(9), pp. 845-851.

ISO/IEC/IEEE, 2011. *42010:2011 - ISO/IEC/IEEE Systems and software engineering -- Architecture description.* s.l.:IEEE Computer Society.

Kallestrup, K. B., Lynge, L. H., Akkerman, R. & Oddsdottir, T. A., 2014. Decision support in hierarchical planning systems: The case of procurement planning in oil refining industries. *Decision Support Systems.*

Kontio, M., 2008. *Architectural manifesto: Adopting agile development, Part 5,* s.l.: s.n.

Kruchten, P., 1995. The 4+1 View Model of Architecture. *IEEE Software,* 12(6), pp. 42-50.

Lothka, R., 2005. *Should all apps be n-tier?*. [Online] [Accessed 12 2014] Available at: http://lhotka.net/weblog/ShouldAllAppsBeNtier.aspx.

Microsoft Patterns & Practices Team, 2009. *Microsoft® Application Architecture Guide (Patterns & Practices).* 2nd ed. s.l.:O'Reilly.

Nguyen, D. et al., 2011. A Methodology for Developing an Agent Systems Reference Architecture. In: *Agent-Oriented Software Engineering XI.* s.l.:Springer Berlin Heidelberg, pp. 177-188.

Norta, A., Grefen, P. & Narendra, N. C., 2014. A reference architecture for managing dynamic inter-organizational business processes. *Data & Knowledge Engineering,* May, Volume 91, pp. 52-89.

Northrop, L., 2003. Chapter 2. What Is Software Architecture?. In: *Software Architecture in Practice.* 2nd ed. Boston, MA: Addison-Wesley.

Object Management Group, 2013. *OMG Unified Modeling Language TM (OMG UML).* 2.5 ed. s.l.:OMG.

Oracle, 2015. [Online] [Accessed 2015] Available at: http://www.oracle.com/us/products/applications/ebusiness/scm/051323.html.

Pääkkönen, P. & Pakkala, D., 2015. Reference Architecture and Classification of Technologies, Products and Services for Big Data Systems. *Big Data Research,* February, 2(4), pp. 166-186.

Stadtler, H., 2005. Supply chain management and advanced planning—basics, overview and challenges. *European Journal of Operational Research,* 163(3), pp. 575-588.

Stadtler, H., 2015. Supply Chain Management: An Overview. In: H. Stadtler, C. Kilger & H. Meyr, eds. *Supply Chain Management and Advanced Planning.* 5th ed. University of Hohenheim: Springer Berlin Heidelberg, pp. 3-28.

Stadtler, H. et al., 2012. *Advanced Planning in Supply Chains. Illustrating the Concepts Using an SAP® APO Case Study.* First Edition ed. Berlin: Springer-Verlag Berlin Heidelberg.

Vidoni, M. & Vecchietti, A., 2015. A systemic approach to define and characterize Advanced Planning Systems (APS). *Computers & Industrial Engineering,* Volume 90, pp. 326-338.

Yonghua Zhou, Yuliu Chen & Huapu Lu, 2004. *UML-based systems integration modeling technique for the design and development of intelligent transportation management system.* s.l., IEEE, pp. 6061-6066.

Zoryk-Schalla, A. J., Fransoo, J. C. & de Kok, T. G., 2004. Modeling the planning process in advanced planning systems. *Information & Management,* 42(1), pp. 75-87.

A Constraint-based Approach for Checking Vertical Inconsistencies between Class and Sequence UML Diagrams

Driss Allaki, Mohamed Dahchour and Abdeslam En-Nouaary
Institut National des Postes et Télécommunications, 2, av ALLal EL Fassi – Madinat AL Irfane, Rabat, Morocco
{d.allaki, dahchour, abdeslam}@inpt.ac.ma

Keywords: UML, Vertical Inconsistencies, Software Development Process, Constraints, Metamodel.

Abstract: The modern software development processes enable evolving software systems and refining models across software life cycle. However, these evolution attitudes may lead to some consistency problems among models at different levels of abstraction. Hence, it is required to discover and detect the potential inconsistencies occurring in models when developing a system. This paper focuses on checking the vertical consistency of UML models using an approach based on defining constraints at the meta-level. These constraints are expressed using EVL (Epsilon Validation Language) to ensure the consistency of models. Representative examples of constraints for checking vertical inconsistencies between class and sequence diagrams are proposed to illustrate our contribution.

1 INTRODUCTION

Over the past few years, modeling systems has long been an essential practice in software development, since a model is supposed to anticipate the results of coding. Indeed, a model is an abstract representation of a system intended for understanding, studying and documenting the system (Cernosek and Naiburg, 2004). Each member of the project team, from the user to the developer, uses and enriches the model differently. Also, the model has the particular advantage of facilitating traceability of the system, namely the possibility of starting from one of its components and monitors its interactions and relationship with other parts of the model.

To illustrate what a model is, *Grady Booch* draws a parallel between a software development and a building construction. This analogy is appropriate since the plots plans to construct a building perfectly reflects the idea of anticipation, design and documentation of the model. However, we note that in building modeling, this anticipation does not take into account the changing needs of users, the starting hypothesis is that these needs are defined once and for all. Yet, in many cases, in software development, these needs change over the project; that is why it is important to manage change and recognize the need to continue supporting our models. Then, unlike what is done in the construction industry, the software modeling process must be adaptive rather than predictive.

From this perspective, a software modeling process defines a sequence of steps, partially ordered, which contribute to the realization of software or changing an existing system (Jacobson et al., 1999). Then, the purpose of a development process is to produce quality software that meets the changing needs of the users in predictable time and cost. To this end, most of modern software modeling processes adopt *iterative* and *incremental* strategies as is the case in *agile* context. The iterative approach is based on the growth and the successive refinement of a system through multiple iterations, feedback and cyclical adjustment being the main engines to converge on a satisfactory system. In the *incremental* development, we split the tasks into small parts, plan them to be developed over time and incorporate them as soon as they are completed. When *agile* modeling is based on some simple principles with common sense that encourage changing models perspectives if needed, and motivate creating multiple models simultaneously.

According to (Huzar et al., 2004), the incremental and iterative nature of software systems and the agile and flexible software modeling processes are one of the main causes of model inconsistencies. An inconsistency roughly means that overlapping elements of different model aspects do not match each other (Allaki et al., 2014). Or in other words, the

Allaki, D., Dahchour, M. and En-Nouaary, A.
A Constraint-based Approach for Checking Vertical Inconsistencies between Class and Sequence UML Diagrams.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 441-447
ISBN: 978-989-758-187-8

whole system is not represented in an harmonized way in different views of its model.

These inconsistencies could be the source of many errors and could therefore invalidate the models and complicate the whole software development process. Especially when adopting a Model Driven Engineering (MDE) approach (Schmidt, 2006). The Object Management Group vision of MDE is called Model Driven Architecture (MDA, 2003). MDA formulates well-established rules and good practices such as adopting the Unified Modeling Language (UML, 2015) as a *de facto* standard for modeling software systems. UML is defined as a graphical and textual modeling language composed of multiple diagrams that unifies both notations and object-oriented concepts. The concepts transmitted by a diagram have a precise semantics and are carriers of meaning. For example, semantics expressed by class and sequence diagrams makes them the most complementarily related diagrams containing meaningful information about both the structure and the behavior of the system being investigated; which makes them also the most refined diagrams during all different software development phases. For this reason, we consider and focus in this work, on examples of inconsistencies between these two diagrams.

In this paper, we first explain, using examples, *how scalable development processes using iterative, incremental and adaptive methods are behind the occurrence of vertical (inter-model) inconsistencies* (i.e. inconsistencies arising among UML model diagrams at different levels of abstraction). After that, we will describe *how our proposed constraint-based consistency checking proposal works*, and we will propose, thereafter, a set of constraints that deal with the given examples of vertical inconsistencies between class and sequence diagrams introduced before.

The rest of the paper is organized as follows. Section II provides three motivating examples of vertical inconsistencies between class and sequence diagrams. Section III presents our constraint-based approach for checking UML model inconsistencies, illustrated by examples dealing with the given vertical inconsistencies, and discussed according to the advantages and limitations of related works.

2 VERTICAL INCONSISTENCIES BETWEEN CLASS AND SEQUENCE DIAGRAMS

The inherent complexity of software systems during their creation will continue to grow as they are evolving, either using traditional or agile software development processes. Indeed, mixing between iterative, incremental and adaptive strategies affects models' consistency by adopting some change attitudes in different development phases. More explicitly, these attitudes advocate assuming models' simplicity, enabling change, using multiple models and so on. The cited attitudes encourage to not over-modeling the system in the first steps of development; which means not depicting additional features in our models until the system requirements evolve in the future. This can be done by developing a small model, or perhaps a high-level model, and evolve it over time (or simply discard it when no longer need it) in an incremental manner. Moreover, we have to use multiple models to develop software, depending on the exact nature of the software we are developing. All these attitudes can lead to numerous conflicts in models across different levels of abstraction. Thus, vertical inconsistencies can arise as a result.

Being aware of this fact, particular attention should concern checking this kind of inconsistencies, as well as others, to undergo changes during a software life cycle, correct errors, accommodate new requirements, and so on.

In what follows, we present some motivating examples from literature that illustrate the conflicts arising between class and sequence diagrams at different levels of abstraction.

Hereafter, we consider that the different parts of the sequence diagrams presented in the following examples are a refinement, at the instance level, of an existing sequence diagram defined in a higher level (specification level). The refinement is used to present more details on the interaction between the objects used in these examples. This lead to assume that the class diagrams are on a higher level of abstraction than the given sequence diagrams.

Example 1: (Connector Type Incompatibility)

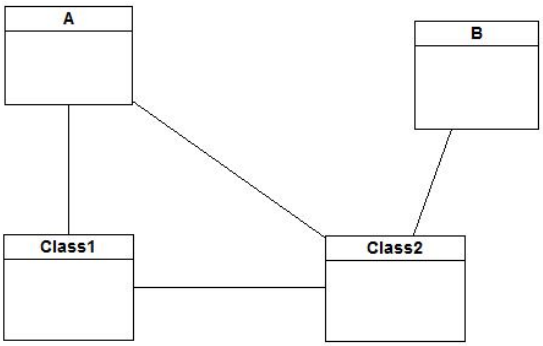


Figure 1: A part of a class diagram (1).

Figure 2: A part of a sequence diagram (1).

The part of sequence diagram illustrated in figure 2 shows an instance of class "A" sending a new introduced message "msg" to an instance of class "B" although there is no direct relationship between the two classes "A" and "B" in the class diagram of figure 1.

If we consider, for example, an incremental context, this inconsistency could occur when we are developing, in the phase of design, a new increment of the software system. For instance, when adding new functionalities to the system, in this new under development increment, we can introduce a new message that links between two objects of two existing classes without updating the class diagram by linking these two classes. Or without editing the sequence diagram in progress to be sure that all messages link only between related classes. This kind of attitudes is common and may appear in an unnoticed way in the context of refining the design of the system being developed following an incremental strategy.

Example 2: (Dangling Operation)

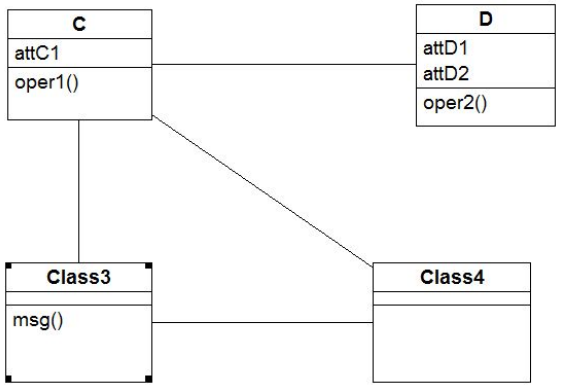


Figure 3: A part of a class diagram (2).

The part of sequence diagram illustrated in figure 4 represents an instance of class "C" sending a new introduced message "msg" to an instance of class

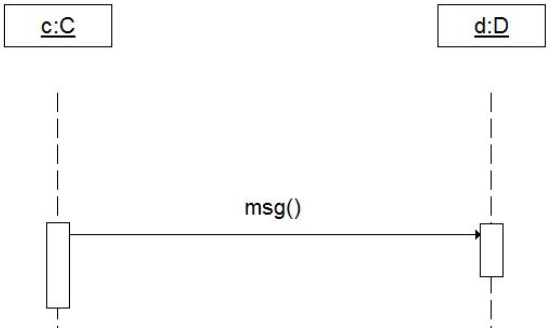


Figure 4: A part of a sequence diagram (2).

"D". However, the message "msg" refers to an operation in class diagram illustrated by figure 3 that does not belong to the class "D" attached to the receiving event of the message in sequence diagram.

When adapting models, for example in an agile software development process, it is common to change design, or part of it, due to the change of initial requirements. This change may lead to some inconsistencies that concern the behavioral aspect of the model. For instance, during these design changes, some operations in the class diagram may not be moved to another class, or sometimes may not be removed from the model. And then, these operations can be referred in a wrong way in the other diagrams; like the case of the *dangling operation* presented before.

Example 3: (Navigation Incompatibility)

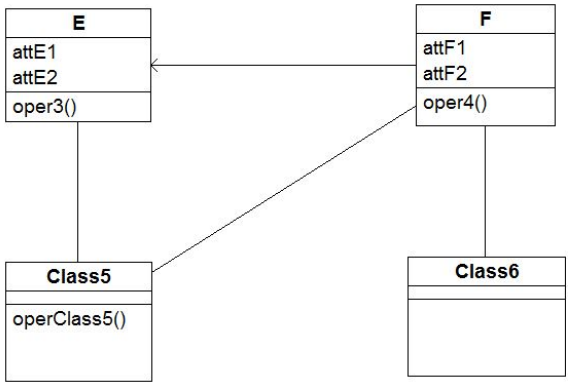


Figure 5: A part of a class diagram (3).

In the part of sequence diagram illustrated in figure 6, a message is sent from a sender object "E" to a receiver object "F" in opposition to the navigation direction of the association between the two corresponding classes "E" and "F" in the class diagram represented in figure 5.

If rearrangements are carried out on an existing part of the system, the example of *navigation incompatibility*

Figure 6: A part of a sequence diagram (3).

could occur. For example, when adopting an iterative strategy in the development process, we develop the least possible before the system is submitted to evaluation. And then, we can neglect, in the first iterations, some details in design; such as the navigation direction of the association between classes. But when refining the model, such information could be added, and then it becomes crucial to adopt the other parts of the model to these changes. Then, for instance, sending a message between two objects without taking into consideration the navigation direction of the association linking between their respective classes is not allowed.

As pointed before, different types of inconsistencies can be encountered in UML models. In this paper, we focus on vertical inconsistencies. The taxonomy presented in (Allaki et al., 2015) proposes more examples and more details about a comprehensive classification of inconsistencies.

3 OUR PROPOSED CONSTRAINT BASED APPROACH

In this section, we present the approach we used for checking the consistency of UML models. Our technique is based on formal constraints defined at the metamodel of UML. These constraints are implemented using EVL (Epsilon Validation Language, 2015) by matching related diagrams' features at the metamodel level.

3.1 An Overview of our Approach

Our EVL constraint-based approach matches UML meta-elements to ensure models' consistency. In our context, the constraints added at the meta-level describe different conditions that UML models have to satisfy to be considered consistent. These conditions concern, syntactically and semantically, the homogeneity, the complementarity and the compatibility of the UML diagrams' elements. Then, checking inconsistencies will be based on detecting violations of consistency according to these constraints. Since the consistency constraints are defined at the UML metamodel level, they have the advantage of being independent from any specific implementation platform and so they can be applied generically to all UML models since any UML model inherits all the specifications, including constraints, from its metamodel.

Note that these constraints will be enabled once the modeler explicitly asks the validation of his model and not during modeling. Thus, some "fake inconsistencies" such as incompleteness or anomalies that are intentionally produced when the model is under construction, could be avoided.

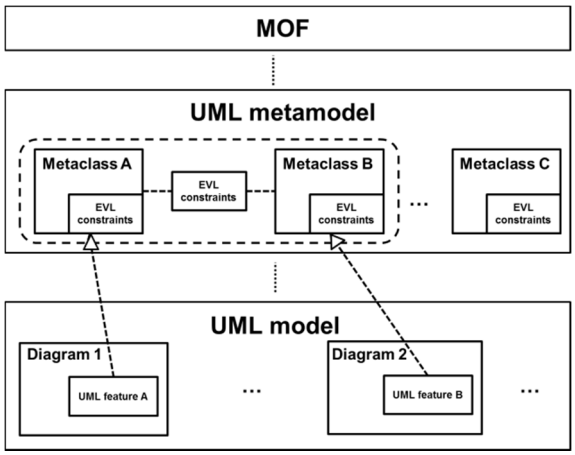


Figure 7: Constraints in UML metamodel level.

On the other hand, recall that UML design models are typically expressed as a large collection of interdependent and partially overlapping UML diagrams. These diagrams relate to different aspects of the system, and are somehow related to each other, as some of their elements have matching links. These links are expressed by the different meta-associations between meta-classes in the UML metamodel.

Our solution exploits these facts to check inconsistencies that can arise between multiple views of the model, even if they are at different levels of abstraction. The key idea behind our approach is a matching between meta-classes by establishing the right links when defining a consistency constraint at UML metamodel. The definition of such constraints is basically done by first, choosing the right meta-classes involved in the constraint, and then, by determining the way these meta-classes are linked.

3.2 Examples of EVL Constraints

In what follows, we produce, for each given example in (Section 2), the UML meta-classes concerned by the inconsistency and the associated constraint expressed in EVL.

EVL (Epsilon Validation Language) is a task-specific language of the general model management language Epsilon (Epsilon, 2015). EVL is a language dedicated to validate models. In their simplest form, constraints expressed in EVL are quite similar to OCL constraints. However, unlike OCL, EVL supports dependencies between constraints (e.g. if constraint A fails, do not evaluate constraint B), supports user interaction (specifies customizable error messages and quick fixes for failed constraints), supports all the usual programming constructs and the convenient first-order logic OCL operations and so on (Kolovos et al., 2015).

All EVL features are suitably integrated in Eclipse Modeling, the CASE tool we used to implement our approach.

Example 1: (Connector Type Incompatibility)
The involved inconsistency elements from the UML metamodel are shown in the following figure.

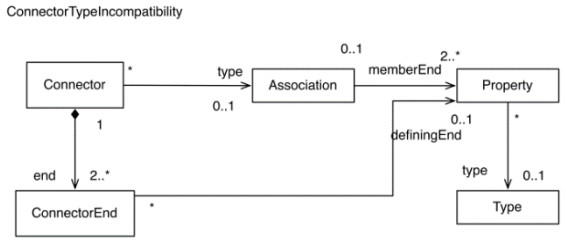


Figure 8: Involved elements from the UML metamodel in the Connector Type Incompatibility inconsistency.

The EVL constraint that checks this inconsistency is presented as follows:

```
    context Connector {

    constraint ConnectorTypeIncompatibility {

    check    :    self.type.memberEnd.type    =
self.end.definingEnd.type

     message : "A model contains a connector"
+ self.name + "for which the type of the
connectable elements that are attached to the
ends of the connector don't conform to the type
of the association ends of the association that
types the connector"
    }
    }
```

In this example, we choose the meta-class *Connector* as a context of the EVL constraint. We make sure if the types of the connectable elements that the ends of the connector are attached conform to the types of the association ends of the association that types the connector. And if this inconsistency appears, a message explaining the situation is displayed.

Example 2: (Dangling Operation)
The part of UML metamodel containing the adequate meta-classes involved in this inconsistency is shown in figure 9.

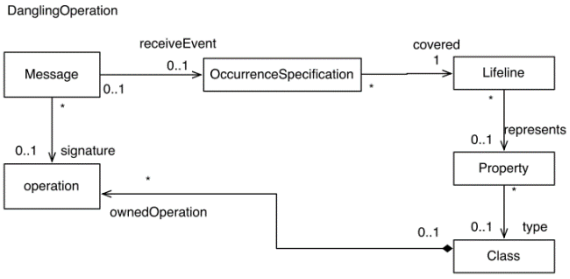


Figure 9: Involved elements from the UML metamodel in the Dangling Operation inconsistency.

In this example, we consider for clarity reasons the simplest instance of the *Dangling Operation* in which we have just one operation in the class. The EVL constraint that checks and fixes this inconsistency is presented as follows:

```
    context Message {

    constraint DanglingOperation{

    check       :       self.signature       =
self.receivedEvent.covered.represents.type.Ow
nedOperation.signature

     message : "A sequence diagram contains a
message" + self.name + " which refers to an
operation that does not belong to the class
attached to the receiving event of the message"

    fix { title : "add an operation to the
class"
          do { var op = new Operation;
               op.name=
self.name;Class.ownedOperation.first().conten
ts.add(op);
          }
           }
    }
    }
```

To deal with the *Dangling Operation* inconsistency, we choose for the corresponding constraint, the meta-class *Message as a* context. The

objective then is to compare the *signature* of the *Operation* referenced by the *Message* with the *signature* of the *Operation* belonging to the C*lass* attached to the receiving event of the *Message* in the Sequence diagram. If the two signatures are different, the inconsistency occurs and therefore a useful message is displayed with a proposition of fixing the inconsistency by creating a new operation to the corresponding class.

Example 3: (Navigation Incompatibility)
The involved meta-classes of this inconsistency are shown in figure 10.

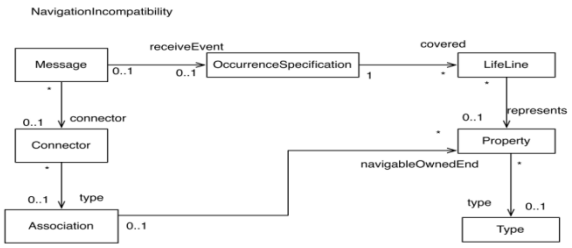


Figure 10: Involved elements from the UML metamodel in the Navigation Incompatibility inconsistency.

The EVL constraint that checks the simplest form of this inconsistency is presented as follows:

```
    context Message {
    constraint NavigationIncompatibility{
    check :
    self.receivedEvent.covered.represents.type
= self.connector.type.navigableOwnedEnd.type
     message : "A sequence diagram contains a
message" + self.name + "of which calling
direction does not match the navigation
constraint on the corresponding association"
    }
    }
```

The context chosen for the *Navigation Incompatibility* constraint is the meta-class *Message*. By defining this constraint, we aim to compare the calling direction of the message if it matches the navigation constraint on the corresponding association. An explanatory message is displayed if the inconsistency arises.

3.3 Discussion

Over the past few years, ensuring consistency in UML models has been a priority investigation for researchers and practitioners in software engineering. As a result, several approaches have been devised to deal with this issue. These approaches can be classified into two categories, namely *transformation-based* techniques and *constraint-based* techniques.

Transformation-based techniques, for example but not limited to (Hanzala and Porres, 2015); (Miloudi et al., 2011), (Straeten et al., 2007) and (Yao and Shatz, 2006) are founded on detecting inconsistencies, after transforming semi-formal UML models to a formal language, using inference mechanisms of that language.

These methods provide us with solid mathematical foundation, proof and tools and add more precision to UML models by avoiding ambiguities when handling inconsistencies in these models.

On the other side, constraint-based techniques, such as (Przigoda et al., 2016), (Kalibatiene et al., 2013), (Sapna and Mohanty, 2007), (Egyed, 2007) and so on, detect inconsistencies in accordance to the formal constraints defined at the metamodel level.

These methods are extensible, by giving the possibility to include new checks for new arising inconsistencies. Also, unlike transformation techniques, they preserve all the information expressed in the UML models; and make the model more expressive through the constraints defined at the metamodel.

However, most of the existing constraint-based proposals generally deal with static aspects of the UML models and are limited to checking inconsistencies in a single diagram, which compromise their efficiency.

Giving the pros and cons of the existing inconsistency checking methods, our proposed constraint-based solution overcomes some of these limitations since it is conceived to ensure the quality and the usefulness of the proposal. Our proposal is easily automated (implemented using Eclipse Modeling). Moreover, EVL, the language used to write constraints, provides much helpful functionality such as the support of quick fixes and the customizable error messages. This can motivate industrial development communities to use it, unlike most of the existing formal techniques that are hard to automate and require a strong mathematical background to apply them. Furthermore, the constraint-based nature of our proposal supports extension mechanisms to deal with any new arising inconsistency. In addition to that, our proposal was designed to be complete in terms of coverage of both potential inconsistencies and the UML diagrams commonly used such as the class, sequence, activity, statechart diagrams and so on; which make it a simple and practical consistency checking proposal.

4 CONCLUSIONS

We tried through this paper to deal with the case of the vertical inconsistencies caused by the refinement of the model. Models are generally refined because of the iterative, incremental and adaptive nature of the modern software development processes.

We explained how our constraint-based consistency checking proposal treats this type of inconsistencies. Our approach adds constraints at the metamodel level by matching the common concepts among the UML diagrams. These constraints, written using the Epsilon Validation Language, automatically help detecting and fixing inconsistencies. To illustrate our approach, we have considered examples of constraints that check vertical inconsistencies arising between class and sequence diagrams.

On the other hand, our proposal is characterized by its ease of automation (implemented using Eclipse Modeling), ability to be extended and completeness of covering all the potential inconsistencies that can affect all the commonly used UML diagrams.

As a future work, we intend to develop a consistency checking process that regroups the best-practices of detecting and handling UML model inconsistencies and that focuses on defining the different steps needed to well behave with the detected inconsistencies. We will apply this on a case study that contains patterns involving a set of tricky examples of inconsistencies and that covers a larger number of expressive UML diagrams. We will also provide further discussion about the experimental results with the Eclipse tool and its performance.

REFERENCES

Cernosek, G., Naiburg, E., 2004. The Value of Modeling.A technical discussion of software modeling. (IBM).

Jacobson, I., Booch, G., Rumbaugh, J., 1999. *Software Development Process*, An Imprint of Addison Wesley Longman, Inc.

Huzar, Z., Kuzniarz, L., Reggio, G., Sourrouille, J.L., 2004. Consistency problems in UML based software development. In UML Modeling Languages and Applications, «UML» 2004 Satellite Activities, Lisbon, Portugal, October 11-15, 2004, Revised Selected Papers. LNCS, vol. 3297, pp. 1-12.

Allaki, D., Dahchour, M., En-nouaary, A, 2014. A New Taxonomy of Inconsistencies in UML Models: Towards Better MDE. In the Proceedings of the 9th International Conference on Intelligent Systems: Theories and Applications, (SITA'14), May 2014, Rabat, Morocco, pp.121-127.

Allaki, D., Dahchour, M., En-nouaary, A, 2015. A New Taxonomy of Inconsistencies in UML Models with their Detection Methods for better MDE. In *International Journal of Computer Science and Applications, Technomathematics Research Foundation*, Vol.12, No.1, pp.48–65.

Schmidt, D, 2006. Guest editor's introduction: Model-Driven Engineering. In IEEE Computer Society, February 2006, Volume 39, No. 2, pp. 25-31.

MDA Guide Version 1.0.1, <http://www.omg.org/mda>, 2003. (Last accessed November 2015).

Unified Modeling Language: Superstructure. Version 2.5, <http://www.omg.org/spec/UML/2.5/>, 2015. (Last accessed November 2015).

Epsilon Validation Language, 2015. <http://www.eclipse.org/epsilon/doc/evl/>, (Last accessed November 2015).

Epsilon, 2015. <http://www.eclipse.org/epsilon/doc/>, (Last accessed November 2015).

Kolovos, D., Rose, L., Domínguez, A.G., Paige, R., 2015. *The epsilon book*. February 4, 2015.

Hanzala, A. K., Porres, I., 2015. Consistency of UML class, object and statechart diagrams using Ontology Reasoners. In *Journal of Visual Languages & Computing.*Volume 26, February 2015, pp. 42–65.

Miloudi, K. E., Amrani, Y. E., Ettouhami, A. 2011. An Automated Translation of UML Class Diagrams into a Formal Specification to Detect UML Inconsistencies. In The Sixth International Conference on Software Engineering Advances, ICSEA 2011, Barcelona, Spain, pp. 432–438.

Straeten, R.V. D., Jonckers, V., Mens, T. 2007.A Formal Approach to Model refactoring and Model refinement. In *Software and System Modeling*, Volume 6, Number 2, June 2007, pp. 139–162.

Yao, S., Shatz, S. M., 2006. Consistency Checking of UML dynamic models based on Petri Net techniques. In 15th International Conference on Computing (CIC 2006), November 21-24, 2006, Mexico City, Mexico, pp. 289–297.

Przigoda, N., Wille, R., Drechsler, R., 2016. Analyzing Inconsistencies in UML/OCL Models. In *Journal of Circuits, Systems and Computers*, Volume 25, Issue 03, March 2016.

Kalibatiene, D., Vasilecas, O., Dubauskaite, R., 2013.Ensuring Consistency in Different IS Models – UML Case Study. In *Baltic Journal of Modern Computing, Volume 1*, No. 1-2, 2013, pp. 63-76.

Sapna, P. G., Mohanty, H., 2007. Ensuring consistency in relational repository of UML models. In 10th International Conference in Information Technology, ICIT 2007, Roukela, India, 17-20 December 2007, pp. 217–222.

Egyed, A., 2007. Fixing inconsistencies in UML design models. In 29th International Conference on Software Engineering (ICSE 2007), Minneapolis, MN, USA, May 20-26, 2007, pp. 292-301.

Challenges and Opportunities in the Software Process Improvement in Small and Medium Enterprises: A Field Study

Gledston Carneiro da Silva and Glauco de Figueiredo Carneiro
Salvador University (UNIFACS), Bahia, Brazil
gledston.silva@pro.unifacs.br, glauco.carneiro@unifacs.br

Keywords: Software Quality, Software Process Improvement, Small and Medium Enterprises, Field Study.

Abstract: The characteristics and profiles of organizations are important issues for the planning of their software process improvement. It supports the alignment with organizational culture as well as with the consolidation of best practices already implemented. This paper presents the results of a field study to identify the perception of the industry about challenges and opportunities of software process improvement faced by Small and Medium Enterprises. This field study aimed at identifying the profile and perception of a group of software development firms concerning software process improvement. The results indicated a list of challenges and activities faced and performed by the companies toward the software process improvement journey.

1 INTRODUCTION

The term *SME* stands for Small and Medium Enterprises and covers a wide range of definitions and measures, varying from country to country and between the sources reporting SME statistics (Ayyagari et al., 2007). Some of the commonly used criteria to identify these companies are the number of employees, total net assets, sales and investment level. However, the most common definition is based on the number of employees. A large number of sources define an SME to have a cut-off range of 0-250 employees (Ayyagari et al., 2007).

Small and medium enterprises (SMEs) are fundamental for the economy of the majority of countries. In countries such as United States, Brazil, Canada, China and India, companies with this profile have significant representativeness in the economy. For this reason, an effective Software Process Improvement (SPI) should take into account difficulties inherent to this type of organization (Dybå, 2005).

This paper presents the results of a field study focused on SMEs to answer the following research questions: i) *What are the main features, challenges and difficulties faced by SMEs that work in software development towards software process improvement?* and ii) *What are the possibilities to support those organizations to overcome these difficulties and to encourage them to embrace the SPI journey?*

The next sections of this paper are organized as follows. Section 2 presents related works. In Section 3 we briefly present the insights acquired during preliminary research, namely when we conducted a Systematic Literature Review (SLR) to find out evidences in the literature about difficulties faced by SMEs during the SPI journey (Silva and Carneiro, 2016). Section 4 discusses the results of a field study to characterize the profile and perception of a set of Brazilian SMEs to compare them with results presented in the SLR (Silva and Carneiro, 2016). Section 5 presents final remarks.

2 RELATED WORKS

In a pilot search for secondary studies using the strings (Systematic Literature Reviews OR SLR) AND (SPI OR software process improvement) on the repositories Digital Library ACM, IEEE Xplore, Science Direct and Google Scholar, we found five Systematic Literature Reviews (SLR) that are somehow related to our two research questions (Pino et al., 2008)(Lavallée and Robillard, 2012)(Bjørnson and Dingsøyr, 2008)(Unterkalmsteiner et al., 2012)(Sulayman and Mendes, 2011).

Pino and colleagues (Pino et al., 2008) published in 2008 a systematic literature review to identify reports and studies focusing on efforts of SMEs in the software process improvement journey. The main goal of their paper was to analyze approaches of SPI related to these type of organizations. According to the same authors, the following items can influence

Silva, G. and Carneiro, G.
Challenges and Opportunities in the Software Process Improvement in Small and Medium Enterprises: A Field Study.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 448-455
ISBN: 978-989-758-187-8

on the success of SPI adoption in SMEs: (a) hire expert advice on software process improvement; (b) acquire financial support to fund the software process improvement; (c) establish cooperation among organizations interested in software process improvement so that they can share resources; (d) perform a gap analysis; (f) establish and institutionalize a communication plan considering all stakeholders; (g) motivate senior management sponsorship and strong commitment of all stakeholders. In this paper, we considered papers published until December 2015 and also difficulties and challenges faced by the companies. Moreover, we compared the obtained results with the ones collected in the field study.

Lavallee and Robillard (Lavallée and Robillard, 2012) published in 2012 a systematic literature review to identify papers that focused on the impact of SPI on developers. Among the positive impacts authors identified the reduction in the number of crises, and increase in team communications and morale, as well as better requirements and documentation. On the other hand, as negative impacts they mention the increased overhead on developers through the need to collect data and compile documentation, an undue focus on technical approaches, and the fact that SPI is oriented toward management and process quality, and not towards developers and product quality. Our work did not consider only the developers perspective, but the perspective of the company as a whole.

In (Unterkalmsteiner et al., 2012), the authors identified and characterized evaluation strategies and measurements used to assess the impact of different SPI initiatives. The systematic literature review conducted by the authors included 148 papers published between 1991 and 2008. Seven distinct evaluation strategies were identified, whereas the most common one, Pre-Post Comparison, was applied in 49% of the inspected papers. Quality was the most measured attribute (62%), followed by Cost (41%) and Schedule (18%). Despite these strategies have also been used to evaulated SPI in SMEs, this SLR did not focused exclusively in these types of companies. Thereby, the authors did not mentioned if these strategies are effective for them. The results of this work corroborate the importance of strategies for the success of software process improvement, and we considered this fact in the analysis of the results of this paper.

According to Google Scholar, among these four systematic reviews, (Bjørnson and Dingsøyr, 2008) had the largest number of citations. It was published in 2008 and reports a systematic literature review of empirical studies of knowledge management in software engineering. Among the selected primary studies, there are three publications (Basri S, 2011)(Baskerville R, 1999)(C.G.v. Wangenheim, 2006) that discuss the use of knowledge management approaches to support software process improvement in SMEs. The focus of selected papers reported in (Bjørnson and Dingsøyr, 2008) is not only on SMEs, however the three studies reveal the importance of these companies in the overall context of SPI (Silva and Carneiro, 2016).

In accordance to the SLR published in 2011 (Sulayman and Mendes, 2011), very few studies (only eight) have specifically focused on SPI for Web companies, despite the large number of existing Web companies worldwide, and the even larger number of Web applications being currently developed. The selected studies did not suggest any customized model or technique to measure the SPI of small and medium Web companies. The measures of success for small and medium Web companies, as per SR results, include development team and client satisfaction, increase in productivity, compliance with standards and overall operational excellence. In addition to the limited number of papers (eight studies), the authors did not addressed difficulties and challenges faced by these companies. In order to be successful in the SPI journey they also need to know beforehand possible pitfalls of such adoption. Moreover, the SLR step of this work was based only on eight papers.

3 INSIGHTS FROM A PREVIOUS SYSTEMATIC LITERATURE REVIEW

This section presents some of the results of a SLR conducted by the authors and published at (Silva and Carneiro, 2016). We aimed to answers the following research question **(RQ)** by conducting a methodological review of existing research: *What are the challenges and difficulties faced by SMEs in the adoption of software process improvement?*

The knowledge of these challenges and difficulties support the SMEs to plan and perform SPI alignment with expectations and organizational culture of these companies.

We conducted the SLR in journals and conferences. We extracted 33 peer-reviewed literature papers published from January 2004 to June 2015 (inclusive). Based on the research question, keywords were extracted and used to search the primary study sources. The search string is presented as follows and used the same strategy cited in (Chen and Babar, 2011): *(challenges OR difficulties) AND (small and medium enterprises OR sme) AND (SPI OR software*

process improvement)

Potentially Relevant Studies. Results obtained from the automatic search and manual search were included on a single spreadsheet: an overall total of 56 results, namely 54 from the automated search plus 02 from the separate manual search. The studies were sorted by title in order to eliminate redundancies. Studies for which the title, author(s), year and abstract were identical were considered redundant. Forty six papers remained after removing the redundant items.

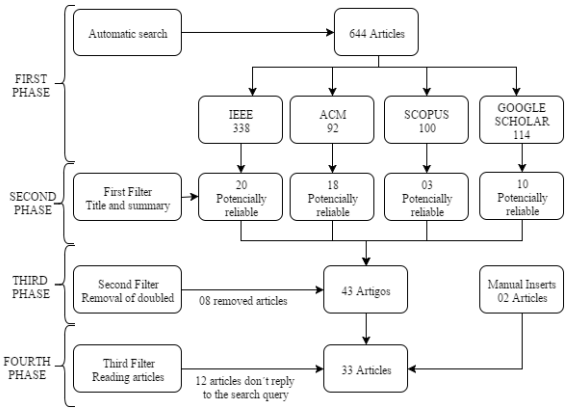


Figure 1: Systematic Literature Review Phases in Numbers (Silva and Carneiro, 2016).

Selection Process. The selection process ended up with 36 papers from which we could obtain evidences to answer the research question. Figure 2 depicts the temporal distribution of the selected studies. As can be observed, more than 97% of the papers were published after 2006, including this year. This is an evidence that the research on software process improvement in small and medium enterprises has been gaining increasing interest from the international software engineering community. At the end, 36 papers were considered relevant to answer the research question **(RQ)**. Table 1 presents an overview of the selection process per public data source (Silva and Carneiro, 2016).

Table 1: Selection Process Overview per Public Data Source (Silva and Carneiro, 2016).

Public Data Source	Search Result	Relevant Studies	Search Effectiveness
IEEE	338	19	5,6%
ACM	92	02	2,1%
SCOPUS	100	03	3%
GOOGLE	114	11	9,6%

We sorted the top ten papers in descending order according to their respective citation number in Google Scholar[1]. The paper M33[2] had the largest number of citations and reports a systematic literature review of empirical studies of knowledge management in software engineering. Among the selected primary studies, there are five publications (M24, M30, M33) that discuss the use of knowledge management approaches to support software process improvement in SMEs. Despite the selected papers of this SLR do not focus only on these types of companies, these studies are evidences of its importance in the overall context of SPI.

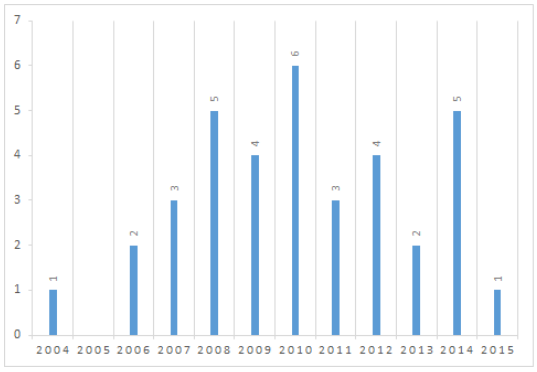


Figure 2: Timeline Distribution of Papers (Silva and Carneiro, 2016).

Characteristics, Challenges and Difficulties faced by SMEs towards SPI Adoption. The selected studies show evidences of peculiarities regarding SMEs, indicating the necessity of specific approaches and solutions for an effective software process improvement adoption. This is especially evident when comparing this scenario with large organizations. Factors related to *financial restrictions and human resource constraints (Challenge 1)* were cited in 24 papers of the 36 of the SLR (M01-M07, M10, M12-M17, M19, M22-M24, M26, M28, M29, M31, M32 and M36), representing 67% of the selected papers (Silva and Carneiro, 2016). It is worth to mention that this does not only refers to direct costs (external consulting and training, for example), but also to indirect costs such as effort required by the team to implement SPI, time required by the teams to understand the SPI rationale and the promote the several required adjustments during the SPI adoption. Due to the financial restrictions, many companies have applied for funding for such end. Typical SMEs project characteristics was the second most cited issue in 14 papers (M01, M03, M04, M07, M11, M12, M14, M15, M17, M19, M20, M27, M28 e M33), i.e. 39% of the selected primary

[1]The list with the top ten papers as well as the selected papers of this SLR are available at http://www.sourceminer.org/slrsme.html.

[2]Also available in the same url indicated above.

studies pointed out this fact. Due to the the ever changing business and customer demands, *projects with short time frame (Challenge 2)* are also a typical characteristic of SMEs, especially for the ones that focus on the evolution of software products of a given domain (Silva and Carneiro, 2016).

The third issue highlighted by the selected studies is *small teams with work overload - (Challenge 3)* (Silva and Carneiro, 2016). This issue was cited by nine papers (M01-M03, M08, M12, M14, M15, M20 and M31), corresponding to 25% of the selected studies. Small teams in SMEs should be taken into account when planning the software process improvement. In this case, participants of the teams can take different roles depending on the demand. There are roles such as quality auditor that can not be assumed by the team members due to conflict of interest. In this case, an external professional can provide an on-demand service for this purpose.

Low focus on process - (Challenge 4) is an evidence provided by eight papers (M01, M03, M08, M12, M14, M15, M19 and M36), corresponding to 22% of the selected papers. It reveals how SMEs deal with daily development practices. This means that organizations can be subjected to conduct software projects in a non-coordinated way and the teams involved in the software projects are not always concerned with software process improvement (Silva and Carneiro, 2016). *Limited number of customers - (Challenge 5)* is another issue pointed out by seven (M01, M03, M06, M07, M12, M15 and M22) (19%) of the selected studies. These type of companies need to maintain the relationship with their clients, otherwise the company can suffer the consequences of the competitors.

Agility to deal with requirements volatility - (Challenge 6) was an issue identified in six papers (M01, M03, M07, M15, M18 and M34) corresponding to 17% of the selected studies. Even after validating the requirements with the client, change in requirements are inevitable. Responding to these demands is crucial to keep clients satisfied. *Absence of training focusing on process - (Challenge 7)* was also identified in six papers (M03, M08, M12, M14, M19 and M35) corresponding to 17% of the selected studies.

Protection of intellectual property - (Challenge 8) also corresponds to 17% of the selected studies (M06, M24, M25, M30, M33 and M35) and has the potentiality to encourage companies and their teams to innovate in new products, new techniques or approaches that can lead to market differential. The next issues are *difficulty to include best practices- (Challenge 9)*, mentioned in five studies (M01, M12, M14, M19 and M36) and *high cost of SPI qualified professionals - Challenge 10* - reported in studies M02, M05, M11, M13 and M34 (Silva and Carneiro, 2016). The results of this SLR were used as reference for the field study presented in the next section.

4 FIELD STUDY IN SMALL AND MEDIUM ENTERPRISES

This sections presents the results of a field study to characterize the perception of Small and Medium Enterprises (SMEs) from Feira de Santana city of State of Bahia in Brazil. The reason for having chosen this city was the perceived number of organizations willing to implement software process improvement practices, despite not having any company officially adopted a well-known software process reference model such as CMMI (Chrissis et al., 2011) or MPS.BR (Montoni et al., 2009). This was considered an appropriate scenario to perform the field study, to collect and analyze data as well as to compare them with the findings obtained in the Systematic Literature Review already conducted by the authors (Silva and Carneiro, 2016) and briefly presented in the previous section.

4.1 Planning and Execution

We contacted 30 companies from Feira de Santana city. Eleven companies agreed to take part in the study. We conducted semi-structured face-to-face interviews with representatives of these companies to collect data for the characterization study. We considered the forms published in (Sulayman et al., 2012) to be adjusted and used in this study[3]. According to the confidentially agreement, the identification of the companies were preserved. The goal of the first form was to collect data related to the interviewed profile representing each company. The second form collected data related to the company. The third form had the goal to evaluate to which extent the company was aligned to the seven items proposed by (Dybå, 2003) and listed as follows: a) *Business orientation*: the extent to which SPI goals and actions are aligned with explicit and implicit business goals and strategies; b) *Involved leadership*: the extent to which leaders at all levels in the organization are genuinely committed to and actively participate in SPI; c) *Employee participation*: the extent to which employees use their

[3]The adjusted forms are available at http://www.sourceminer.org/slrsme.html.

knowledge and experience to decide, act, and take responsibility for SPI; d) *Concern for measurement*: the extent to which the software organization collects and utilizes quality data to guide and assess the effects of SPI activities; e) *Exploitation*: the extent to which the software organization is engaged in the exploitation of existing knowledge; f) *Exploration*: the extent to which the software organization is engaged in the exploration of new knowledge (Dybå, 2003). Finally, the fourth form aimed at identifying what the organization had already implemented in terms of software process improvement.

Profiles of the Participants. Nine of the participants had worked at least three years in the company. On the other hand, four of the interviewed had worked at least five years. Nine of the interviewed were project managers and the other two took were members of the software engineering process group.

Profiles of the Organizations. Five companies had developed software for at least five years, whereas three companies had operated in the market for at least fifteen years, two companies had between five and ten years. Finally, just one company had less than five years. Considering that eight companies had up to five professionals, one company had seven professionals and two companies had more than ten professionals, we can conclude that there is to some extent a relationship with Challenge 1 reported in the previous section - *financial restrictions and human resource constraints* and Challenge 3 - *small teams with work overload*.

All the companies are located in Feira de Santana city in the Bahia State of Brazil. Due to the majority of companies with less than five employees, the Software Quality process activities, including the audits, can be performed by an external professional. This leads to the possibility of sharing this professional among the companies. This is a viable solution to circumvent the challenge of human resources constraints. This number of professionals is an issue that should certainly influence the occurrences of *small teams with work overload - (Challenge 3)*. This overload can impact the participation of professionals in activities related to the SPI such as process elaboration and review as well as training. And participating in these activities compromise the capacity of the company in its ongoing projects. This has as a natural consequence in the increase of natural expenditures.

Regarding the project duration, 55% of the companies reported that their projects end in a maximum of ten weeks. This is a clear evidence of focus on small projects. In fact, even working with software product evolution, these companies tend to split the activities in small chunks. This was identified in (Silva and Carneiro, 2016) as *projects with short time frame (Challenge 2)*. A possible explanation for this scenario is that the majority of these companies evolve software products for a specific domain. Another fact that was identified during the interviews was that these companies suffer the influence of agile methods, especially Scrum. These approaches somehow motivate teams to deal with small projects.

4.2 Success Factors in SPI Initiatives

Table 2 presents data related to the third form of the field study focusing on *Business Orientation*. Data collected from the field study reveal which items from the *Business Orientation* were considered relevant by the companies: 1 - *We have established unambiguous goals for the organizations SPI activities*, was mentioned by 64% of the companies. Companies developed their process based on strategic planning and directed/adjusted efforts of SPI aligned with their needs. Approximately 64% of the companies informed that *our SPI goals are closely aligned with the organizations business goals*. The establishment of goals can have as a consequence a possible push in business as a result of SPI initiatives.

Table 2: Success Factors in the Business Orientation Perspective.

Activity	Practice
1 - We have established unambiguous goals for the organizations SPI activities.	64%
2 - There is a broad understanding of SPI goals and policy within our organization.	28%
3 - Our SPI activities are closely integrated with software development activities.	55%
4 Our SPI goals are closely aligned with the organizations business goals.	64%
5 We have a fine balance between short-term and long-term SPI goals.	09%

Table 3 describes quantitatively how companies perceive success factors from the Leadership Engagement perspective. A percentage of 73% of the companies inform that the staff actively support SPI activities. As a natural consequence, the activity *8 Management considers SPI as a way to increase competitive advantage* is recognized by 64% of the companies to justify investiment in SPI to improve the quality of software products. On the other hand, 73% of the companies agree that activity *9 Management is ac-*

Table 3: Success Factors in the Leadership Engagement Perspective.

Activity	Practice
6 Management is actively supporting SPI activities.	73%
7 Management accepts responsibility for SPI.	55%
8 Management considers SPI as a way to increase competitive advantage.	64%
9 Management is actively participating in SPI activities.	73%
10 SPI issues are often discussed in top management meetings.	45%

tively participating in SPI activities is associated with the initiatives they have seen in their companies.

Table 4 describes quantitatively how companies perceive success factors from the Employee Participation perspective. Activities 11 (*Software developers are involved to a great extent in decisions about the implementation of their own work*) and *12 - Software developers are actively contributing with SPI proposals* were recognized by 73% of the interviewed companies. In that sense, employees recognize that they are encouraged to suggest changes in projects and process in the company. Despite being initially not confident to provide suggestions, later on they contribute with suggestions related to the SPI activities. This is corroborated by the fact that 91% of the interviewed agreed that *13 Software developers are actively involved in creating routines and procedures for software development*.

Table 4: Success Factors (Software Process Improvement) Employee Participation.

Activity	Practice
11 Software developers are involved to a great extent in decisions about the implementation of their own work.	73%
12 Software developers are actively contributing with SPI proposals.	73%
13 Softwaredevelopers are actively involved in creating routines and procedures for software development.	91%
14 We have an ongoing dialogue and discussion about software development.	64%
15 Software developers have responsibility related to the organizations SPI activities.	64%
16 Software developers are actively involved in setting goals for our SPI activities.	64%
17 We have an ongoing dialogue and discussion about. SPI.	64%

Table 5: Success Factors (Software Process Improvement) Concern for Measurement.

Activity	Practice
18 We consider it important to measure organizational performance.	64%
19 We regularly collect quality data (e.g. defects, timeliness) from our projects.	64%
20 Information on quality data is readily available to software developers.	55%
21 Information on quality data is readily available to management.	55%
22 We use quality data as a basis for SPI.	55%
23 Our software projects get regular feedback on their performance.	64%

Table 5 describes quantitatively how companies perceive success factors from the Concern for Measurement perspective. The majority of the companies (64%) recognize the relevance of measuring organizational performance as stated by activity 18 (*We consider it important to measure organizational performance.* Software life-cycle activities need to be measured to evaluate its performance and indicators are required for this end. Another two activities related to the measurement perspective were also classified as relevant by 64% of the companies: *19 - We regularly collect quality data (e.g. defects, timeliness) from our projects* and *23 - Our software projects get regular feedback on their performance*. The companies reported that indicators were planned, collected an later analyzed/compared with their respective targets. However, they also recognized that there is the need to improve the way measurement is performed during software projects life-cycle, especially due to the effort required to accomplish measurement related activities.

Table 6 portrays success factors regarding Exploitation of Existing Knowledge. The following activities were recognized by 73% of the companies 25 *We are systematically learning from the experience of prior projects* and 26 *26 - Our routines for software development are based on experience from prior projects*. Table 7 focuses on success factors regarding Exploitation of New Knowledge. The activities 31 *In our organization, we encourage innovation and creativity*, 34 *We have the ability to question established truths* had both 82% of representation by

Table 6: Success Factors (Software Process Improvement) Exploitation of Existing Knowledge.

Activity	Practice
24 We exploit the existing organizational knowledge to the utmost extent.	45%
25 We are systematically learning from the experience of prior projects.	73%
26 Our routines for software development are based on experience from prior projects.	73%
27 We collect and classify experience from prior projects.	36%
28 We put great emphasis on internal transfer of positive and negative experience.	55%
29 To the extent we can avoid it, we do not take risks by experimenting with new ways of working.	55%

Table 7: Success Factors (Software Process Improvement) Exploitation of New Knowledge.

Activity	Practice
30 We are very capable at managing uncertainty in the organizations environment.	36%
31 In our organization, we encourage innovation and creativity.	82%
32 We often carry out trials with new software engineering methods and tools.	45%
33 We often conduct experiments with new ways of working with software development.	27%
34 We have the ability to question established truths.	82%
35 We are very flexible in the way we carry out our work.	73%
36 We do not specify work processes more than what are absolutely necessary.	28%

the participants. They reported that their companies have encouraged professionals to propose solutions creatively considering the possibility of rewards.

Finally, the activity 35 *We are very flexible in the way we carry out our work* was confirmed by 73% of the participants that commented that flexibility is a key factor to face the challenges during software project development and evolution in the sense that requirements evolve continuously and the software product must meet the expectations of clients.

4.3 Software Process Improvement in Practice

Table 8 presents the results related to process and activities that have actually been implemented in the companies. Table 9 presents the results of the perception of the companies that took part in the field study regarding the relevance of Software Process Improvement for their business.

Table 8: To what extent the SPI processes/activities listed in the table below are performed in your organization for projects?

Activity	Practice
37 Motivation for the use of CMMI and MPS.BR	73%
38 Requirements Engineering	55%
39 Project Management	64%

Table 9: How important do you think are the following SPI processes/activities for projects in your organization?

Activity	Practice
40 Motivation for the use of CMMI and MPS.BR	73%
41 Requirements Engineering	73%
42 Project Management	64%

An important finding reported in Tables 8 and 9 was despite the relatively high motivation reported by the companies to implement well-known reference models such as CMMI (Chrissis et al., 2011) or MPS.BR (Montoni et al., 2009), they have not yet prepared for the evaluation. Among the reasons provided was *financial restrictions and human resource constraints (Challenge 1)* of the SLR presented in Section 3. The companies in fact revealed that they need external sponsorship to implement such models to overcome direct and indirect costs of this initiative. However, despite these constraints, the companies demonstrate motivation (Table 9) for the challenge of SPI, considering both the interviewed professionals and staff, what is a positive fact to achieve effective results in this journey.

Limitations of the Characterization Study. Two possible limitations were identified in this characterization. The first was related to the fact that the profile, number and location of the companies that took part in the study could somehow prevent the generalization of the results. However, it was verified that the profile of the companies in terms of number of professionals, type, duration and number of software projects and gross operational income were compatible with the values provided by several references of Small and Medium Enterprises that work in the

area of software development. The second limitation refers to the number of companies in the characterization. They correspond to approximately half of the companies of the region, for that reason, for some extent they represent a reasonable sampling in terms of profile. The fact that the majority of the interviewed professionals (82%) was from staff could had influence the answers. In a new version of the study we will select participants from different roles for a better distribution of profiles. However, despite these limitations, we could confirm some of the results from our systematic literature review such as financial restrictions and human resource constraints, projects with short time frame and small teams with work overload.

5 CONCLUSION

This paper presents a characterization of Small and Medium Enterprises aimed at identifying challenges, difficulties and opportunities in the context of Software Process Improvement (SPI). The characterization consisted in comparing the results obtained from a systematic literature review and from a field study. The results present a list of challenges and activities faced and performed by the companies toward the software process improvement journey. They can be a reference for companies that plan to adopt SPI and researchers that can conduct new studies to compare this scenario with other small and medium enterprises experiences.

REFERENCES

Ayyagari, M., Beck, T., and Demirguc-Kunt, A. (2007). Small and medium enterprises across the globe. *Small Business Economics*, 29(4):415–434.

Baskerville R, P.-H. J. (1999). Knowledge capability and maturity in software management. 30(2):26–46.

Basri S, O. R. (2011). Towards an understanding of software development process knowledge in very small companies. *In Informatics Engineering and Information Science*, pages 62–71.

Bjørnson, F. O. and Dingsøyr, T. (2008). Knowledge management in software engineering: A systematic review of studied concepts, findings and research methods used. *Information and Software Technology*, 50(11):1055–1068.

C.G.v. Wangenheim, S. Weber, J. H. G. T. (2006). Experiences on establishing software processes in small companies. *Information and Software Technology*, 48(9):890–900.

Chen, L. and Babar, M. A. (2011). A systematic review of evaluation of variability management approaches in software product lines. *Information and Software Technology*, 53(4):344–362.

Chrissis, M. B., Konrad, M., and Shrum, S. (2011). *CMMI for development: guidelines for process integration and product improvement*. Pearson Education.

Dybå, T. (2003). Factors of software process improvement success in small and large organizations: an empirical study in the scandinavian context. *ACM SIGSOFT Software Engineering Notes*, 28(5):148–157.

Dybå, T. (2005). An empirical investigation of the key factors for success in software process improvement. *Software Engineering, IEEE Transactions on*, 31(5):410–424.

Lavallée, M. and Robillard, P. N. (2012). The impacts of software process improvement on developers: A systematic review. In *Proceedings of the 34th International Conference on Software Engineering*, pages 113–122. IEEE Press.

Montoni, M. A., Rocha, A. R., and Weber, K. C. (2009). Mps. br: a successful program for software process improvement in brazil. *Software Process: Improvement and Practice*, 14(5):289–300.

Pino, F. J., García, F., and Piattini, M. (2008). Software process improvement in small and medium software enterprises: a systematic review. *Software Quality Journal*, 16(2):237–261.

Silva, G. and Carneiro, G. (2016). Software process improvement in small and medium enterprises: A systematic literature review. In *Information Technology-New Generations (ITNG), 2016 13th International Conference on*, pages 552–557. Springer.

Sulayman, M. and Mendes, E. (2011). An extended systematic review of software process improvement in small and medium web companies. In *Evaluation & Assessment in Software Engineering (EASE 2011), 15th Annual Conference on*, pages 134–143. IET.

Sulayman, M., Urquhart, C., Mendes, E., and Seidel, S. (2012). Software process improvement success factors for small and medium web companies: A qualitative study. *Information and Software Technology*, 54(5):479–500.

Unterkalmsteiner, M., Gorschek, T., Islam, A., Cheng, C. K., Permadi, R. B., and Feldt, R. (2012). Evaluation and measurement of software process improvementa systematic literature review. *Software Engineering, IEEE Transactions on*, 38(2):398–424.

Validating Sociotechnical Systems' Requirements through Immersion

Andreas Gregoriades[1] and Maria Pampaka[2]
[1]*Cyprus University of Technology, Limassol, Cyprus*
[2]*The University of Manchester, Manchester, U.K.*
andreas.gregoriades@cut.ac.cy, maria.pampaka@manchester.ac.uk

Keywords: Requirements Validation, Simulation, Human Factors, Virtual Reality.

Abstract: One of the most critical phases in complex socio-technical system development is the validation of non-functional requirements (NFR). During this phase, system designers need to verify that the proposed system's NFRs will be satisfied. A special type of NFRs which is often ignored regards the Human Factors (HF) NFRs. These requirements are of vital importance to socio-technical systems since they affect the safety and reliability of human agents within such systems. This paper presents a scenario-based approach for validating HF NFRs using VR CAVE simulation. A case study is used to demonstrate the application of the method in the validation of the situation awareness NFR of an in-vehicle Smart driver assistive technology (SDAT). Such systems aim to alleviate accident risks by improving the driver's situation awareness by drawing their attention on critical information cues that improve decision making. The assessment of the HF NFR is achieved through an experiment with users in a virtual environment. This work describes and demonstrates a method that utilizes a custom-made, modular 3D simulator that uses a number of hazardous scenarios, for the validation of the HF NFRs of prospective systems.

1 INTRODUCTION

Requirements validation constitutes an important facet of a successful system development. Unlike functional requirements, which can be deterministically validated, non-functional requirements (NFRs) are considered as soft/latent variables not directly observed or implemented; instead, they are satisfied (Zhu et al., 2012) by functional requirements. Despite their importance, NFRs are usually addressed at a late stage of system development, whilst functional requirements are considered at the early phase of software development (Marew, 2009). Therefore, the initial stages of a system's specification may not address the NFRs adequately, which could lead to system failure once the system has been commissioned (Adams et al., 2015). NFR analysis approaches range from unstructured and informal, to highly formal and mathematically-driven. The former include approaches such as KAOS (Nwokeji et al., 2014), a goal-oriented software requirements capturing approach. In the same vein, i* approach (Chung et al., 2000) uses goals and enables the quantification of requirements from goal diagrams. The latter category includes formal methods based on model checking such as Z, Markov, and queuing models (Matoussi, 2008).

This paper introduces a Human Factors (HF) requirements validation method that exploits scenario-based testing through immersion. The application of the methodology is demonstrated through a case study on the analysis of the situation awareness NFR of a future smart driver assistive technology (SDAT). The uses a custom made virtual reality (VR) simulator that mimics the environment and models prototype SDATs using 3D visualizations that simulate the candidate designs.

The paper next reviews the literature in NFR assessment, HF and situation awareness (SA). This is followed by the NFR validation methodology. Next a case study demonstrates the application of the HF NFR validation method, followed by analysis of the data from the experiment and presentation of the emerging results. The paper concludes with a brief discussion of methodological and substantial implications.

2 LITERATURE REVIEW

The majority of NFRs in complex socio-technical systems address system properties such as performance, reliability and security. However, there is an additional dimension that needs to be analysed,

Gregoriades, A. and Pampaka, M.
Validating Sociotechnical Systems' Requirements through Immersion.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 456-463
ISBN: 978-989-758-187-8

which is the human dimension. By definition a socio-technical system exhibits both technical and social complexity. These systems are composed of human and machine entities that work together to accomplish a common goal. Transportation systems belong to this category of complex systems since they incorporate vehicles, drivers, road infrastructure and intelligent systems in vehicles. The technical aspects of these in-vehicle systems refer to the functional requirements of machine agents and the human-machine interaction metaphors. The social facet of the system relates to human factors and the associated human performance constraints. Thus, designing such complex systems requires the investigation of all facets. The technical dimension is addressed by the functional requirements and the system NFRs, while the human dimension is influenced by HF NFRs. These are defined by human agent limitations affected by the diverse nature of human characteristics, such as ability, stress, concentration, SA etc. However, despite their importance as a critical cause of systems failure, human factors have not been adequately considered by practitioners during the design, development, and testing of systems (Gregoriades, 2004).

Moreover, even though human factors and requirements have a lot to share, only a few studies apply human factors knowledge to requirements engineering. While NFR such as performance, security and maintainability are considered for software functions, NFRs for people, such as SA and workload, have received less attention. Such requirements have been proven very significant in preventing system failure, articulated in the form of accidents in complex systems such as transportation (Gregoriades, 2010). Therefore, the systematic analysis of this type of NFRs prior to any system implementation is considered vital. The main problem in validating these requirements is the need for a detailed specification of the envisioned system or the implementation of a prototype system. Both of these activities are time consuming and expensive. The former requires formal methods which are hard to comprehend by stakeholders and the latter requires time effort and cost. Once either of the two is realised it is possible to perform an analysis of system behaviour under a number of test scenarios. Formal methods, though, suffer from being too specific, hence their application in validating NFRs is constrained. Prototyping, on the other hand, provides a more generic model based on which different facets of the system can be tested such as people, technology and tasks. This, however, is expensive and risky. Therefore, the use of a simulated environment for requirements analysis saves the costs of prototypes, especially for complex systems (Sutcliffe et al., 2004) and makes the process safe. This approach, employed in this study, enables testing technological solutions and the evaluation of their effect prior to implementation.

Designing complex systems such as the smart in-vehicle information systems requires the effective and efficient management of requirements. The inappropriate specification of functional and non-functional requirements increase dramatically the risk of failing to meet customer needs (Peng, 2012). Functional requirements have received much attention in this process, while, NFRs have been more or less deliberately ignored (Illa, 2000). This led to a lot of systems failing due to improper management of NFRs. Past research addressed extensively different sets of NFRs along with frameworks of NFR such as Softgoal Interdependency Graphs (SIGs) (Zhu, 2012).

NFR validation has attracted significant attention in recent years due to the importance of NFRs in overall system acceptance. Traditional approaches to NFR validation include prototyping and inspection. Recent approaches focus on the quantitative analysis of NFRs. In our previous work we used a Bayesian Networks (BN) approach to model NFRs using knowledge elicited from the domain (Gregoriades, 2005). NFRs are assessed based on a scenario generation and evaluation algorithm that runs the BN with different input. The output is a quantitative estimation of the satisfiability of the NFR. Other groups (Zayaraz et al., 2005; Sadana et al., 2007) also used a quantitative model to analyze conflicts among NFRs. This approach, however, is limited to high level architectural requirements. In the same vein, Marew and colleagues (Marew et al., 2009) used Quantified Softgoal Interdependency Graphs (QSIGs) to assess the degree of softgoal satisfaction. However, the assessment of QSIGs is based on subjective estimates of the degree of interdependencies among softgoals. Similarly, Zhu et al. (2012) apply fuzzy qualitative and quantitative softgoal interdependency graphs for NFRs tradeoff analysis. Based on the above, it is evident that NFRs assessment is an ongoing research issue. The growing ubiquity of complex sociotechnical systems led to more NFRs to be analysed during systems' design phase. One example of such NFRs is safety which is addressed in this study and expressed in terms of accidents.

2.1 Human Factors & Requirements

NFRs such as performance and maintainability are specified for software or hardware systems. NFRs for

people, such as SA and workload, have received less attention. These requirements, however, have been proven very crucial in preventing system failure. Specifically, in transportation, road accidents are usually attributed to human error (Fuller, 2002; Theeuwes et al., 2012) that is induced from low SA caused by increased workload. Humans, as information processing systems, have a number of information flow channels (visual, auditory, tactile) processing various information sources (e.g. a navigation system display, the forward view through the windscreen) of varied bandwidths (e.g. high-density traffic will require a higher sampling rate than low-density traffic). Our cognitive capacity is limited, and consequently there is an upper threshold to the amount of information we can process per second and channel (Endlsey, 2000; Fuller, 2002; Holohan et al., 1978). Therefore, we tend to share our attention among a few information sources. An overloaded driver is less likely to deal effectively with an unexpected event (Konstantopoulos et al., 2010). Fuller (2012) also expresses accident risk as a function of the driver's cognitive resources and task-demand in the driver-road system.

Therefore, the systematic analysis of these HF NFRs prior to any system implementation is considered vital. The main problem in evaluating these requirements is the need to implement a prototype design of a hardware-software system, which is expensive (Stone et al., 2001). Hence, the use of virtual reality (VR) settings is becoming very popular. One of the most important applications of VR technology has been the use of virtual prototypes for functional requirements analysis (Sutcliffe et al., 2004). However, the use of VR for HF requirements analysis has not been addressed. Essentially, HF requirements can be expressed in terms of a threshold value that defines their minimum quantification or satisfaction level. These define the cognitive and physical capabilities of humans. These capabilities are put to the test when processing dynamically changing information during driving. If these capabilities are reached then this in effect increases the likelihood of committing an error due to high workload. Workload, however, is directly related to SA; the link between the two has been previously established (Gregoriades et al., 2007). When the perceived information increases people tend to prioritise which increases the risk of an incorrect comprehension. In traffic safety, SA constitutes a major critical factor, since it provides the driver with the ability to anticipate events given perceived driving and environmental conditions.

Validating HF requirements for such systems makes the use of VR simulators inevitable due to the complexity, effort and cost associated with the development of prototypes. In the same vein, controlling infrastructural parameters in the real world is unethical. Moreover, ruling out confounding effects to examine the influence of control measures on HF is very difficult in field experiments. Driving simulators provide the researcher with a powerful tool to test driving behaviour under controlled settings. Apart from the usually high cost of the simulator, outsourcing of experiments to analyse driving behaviour using native users is difficult, if not impossible in some cases, due to the large number of subjects needed for reliable results. On the other hand, low cost driving simulators do not provide a sufficient level of realism to analyse human factors. Unrealistic conditions may affect the driving behaviour which effectively could influence the validity of the experimental study. The method proposed herein demonstrates the design of a driving simulator that exploits 3D modelling tools in a scenario-based approach to promote realism and interactive representation of road networks. The approach simplifies the process of implementing 3D road infrastructure models through the utilization of reusable modules that represent different in-vehicle technologies or infrastructural components. This simplifies the process of designing/modifying the simulation model by reusing model constructs in a plug and play fashion, which enables the analyst to easily design a range of experimental conditions (i.e. scenarios), to evaluate assumptions and hypotheses from different perspectives.

2.2 Situation Awareness

SA constitutes a major critical factor in complex socio-technical systems. In transportation, it provides the driver with the ability to anticipate events given perceived driving and environmental conditions. SA defines the process of perceiving information from the environment (level 1), comprehending its meaning (level 2) and projecting it into the future (level 3). SDAT have been developed to alleviate accident risk by either reducing driver workload or assessing driver attentiveness. Examples include adaptive cruise control, collision notification, driver monitoring, traffic signal recognition, night vision, lane departure warning systems and blind spot monitoring. Such systems aim to draw drivers' attention on critical cues that improve their decision making. However, they only provide limited support to SA since they address isolated factors and in some cases with negative effect due to the extra information

load they incur to the driver. The first step in improving drivers' SA is to enhance their capability of perceiving and interpreting traffic and environmental conditions (i.e. level 1 and 2 of the aforementioned SA model). However, such smart systems facilitate level 3 SA for navigation, which might decrease drivers' attention, due to secondary task execution, that could lead to reduced level 1 SA. This could undermine attention to operational or tactical driving activities (e.g. braking, lane changing, gap acceptance etc.). To that end, three important issues need to be addressed prior to any SDAT development: (i) identification of drivers' information needs that could enhance SA, (ii) the specification of a SDAT feedback metaphor (feedback type and appropriate time for issuing warnings) to support those needs without impairing driver attention, and (iii) the evaluation of the effect of a prospective SDAT on traffic safety. This is a complex process and in most cases is only feasible once a prototype of the system is available.

Endlsey et al. (2012) warn socio-technical system designers of the importance of maintaining SA in complex systems and draw attention on the issues that could inhibit SA. One of the most important strains of SA is information overload. Too much information at any point in time hinders human operators' adequate SA. Overloading divides the decision maker's attention among numerous stimuli resulting in increased demand for cognitive resources. This is known as attentional tunnelling (Endlsey, 2012) and results in reduced information scanning capability.

3 NFR VALIDATION METHOD

The proposed NFR validation method is based on the design science (Hevner et al., 2010) paradigm, and in particular its evaluation phase which investigates the effectiveness of an artefact and guides its re-design through changes in specification. Design science synthesises the sciences of the artificial, engineering design, information systems development, system development as a research methodology, and executive information system design theory for the building and evaluating of IT artefacts for specific problems (Hevner et al., 2010). The design and development of new artefacts such as the SDAT, described herein, requires a systematic approach towards artefact design, development and evaluation. This aims to assure that the artefact contributes towards resolving a particular problem.

The method is composed of a number of steps that are executed both in sequence and in parallel at certain stages. Initially, the problem needs to be expressed in terms of human factors specification. This could be articulated in terms of human performance and human reliability, and in particular, as the acceptable SA and workload levels of human agents in a system. These are conditions that could incur high likelihood of human error (Gregoriades, 2010). Once the problem to be analysed is clearly stated and the critical HF NFRs are identified, then the minimum level NFR satisfiability needs to be set. The refinement of HF NFRs into functional specifications which when realised will guarantee the satisfaction of the NFR comes next. This is achieved using a combination of domain knowledge and input from subject matter experts. For instance, guidelines for enhanced SA, as specified by Endlsey (2012), are expressed in terms of information requirements, visualisation metaphors and interaction styles which are functional requirements that the SDAT should have. The next step in the process is the specification of the test scenarios, based on which the artefact is going to be evaluated. Grounded within the problem to be analysed, the goals of the desired virtual environment are set. Accordingly, specifications of the virtual environment to be used for the evaluation of the artefact are also set. During this stage a generic VR simulator is customized based on the above goals, to model the problems in question. The customization of the simulator is composed of three steps: 1) the development of the test environment in terms of buildings, infrastructure and traffic conditions. 2) The modelling of the scenarios, as described by the domain experts; these include atypical events in the simulation that would stress test the subjects in the experiment. 3) The modelling of the virtual version of the artefacts under scrutiny. Prior to its use, the VR simulator needs to be validated against a number of factors such as realism, to guarantee the correctness of the NFR assessment. NFRs quantification is achieved through an experiment with users in the VR environment. The specification of the experiment is defined by an HF expert. The assessment of NFR is then refined into phenotype behaviours that can be monitored in a driving simulator. Phenotype driving behaviours are monitored and logged into the systems database. The logged observations from the simulation are pre-processed, analysed and subsequently collated into a single metric that corresponds to the assessed NFR. The NFR assessment is compared against the desired NFR level. If the minimum level of NFR is not satisfied then the virtual artefact under scrutiny needs to be redesigned. The process is repeated until the NFR is satisfied.

4 CASE STUDY

To demonstrate the application of the method, a case study was conducted for the validation of the SA NFRs of a future SDAT. The NFR evaluation method is based on the paradigm of scenario-based testing. In each scenario, participants were required to drive through a pre-specified path on a road network. Throughout the driving task, participants had to respond to emerging hazardous situations. Situational cues were visualised through the SDAT in the form of a virtual augmented reality head-up display (HUD) interface within the virtual vehicle. The SDAT interface was designed based on identified driver information requirements and domain knowledge (Endlsey, 2012). SDAT designs aimed to address drivers' information needs for better SA. Specifically, vehicle's peripheral traffic, road works, road signs and approaching traffic jam were projected through the virtual SDAT. The goal was to assess the effect of each SDAT design on drivers' SA. Satisfiability of SA NFR is specified as an improvement in drivers' SA using SDAT compared to no SDAT use, and is specified as a threshold value. Two SDAT designs were developed using Endlsey's (2012) design principles for SA support. The functional requirements of the SDAT systems have been implemented using the guidelines of: information prioritization, timeliness and relevance of information, information filtering, familiarity of the visual metaphors, and presentation of information in the right context. These aim to alleviate information overload, reduce display density, enhance driver's ability to comprehend the meaning of information and finally assist in developing projections of the situation into the future. The SDATs utilise the above through fusion of vast amount of information from the environment into meaningful attentional directives/cues that describe the driving situation in real-time.

Figure 1: The driving simulator in the VR CAVE. A participant doing the experiment while being observed by researchers.

As part of the NFR validation method, the first step is the design and implementation of the driving simulator. Figure 1 illustrates the developed simulator in VR CAVE that enables the stereoscopic interaction of participants with the experimental conditions.

Participants are immersed with the experimental scenarios through a combination of augmented reality and tangible interaction styles, for a more realistic experience. The second step in the method is the design of the virtual prototype SDAT systems in the virtual environment. The development of the virtual SDATs is realized using a scripting language. The virtual SDAT had to abide to the functional requirements specified in previous steps. The third step is the specification of the hazardous scenarios.

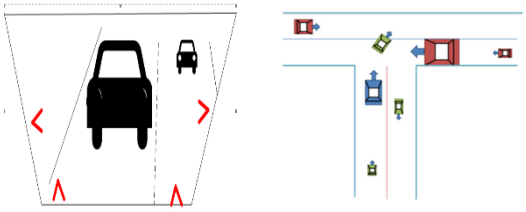


Figure 2: The radar design (right) and information prioritisation –arrow design (left).

The user interface of SDAT systems is of paramount importance in improving SA. Hence, it was designed to provide blind-spot information and to alert drivers of unseen imminent threats. The system uses a combination of HUD with augmented reality capabilities, so that the direction of the threat is clearly comprehended by drivers. The information architecture of the UI aimed to provide the driver with enhanced peripheral vision with a dynamic assessment of the most critical entities within the immediate periphery of the vehicle. The blue print designs of the candidate systems are depicted in Figure 2. In the first design (radar), the host vehicle is shown in a circle (in blue) surrounded by red and green vehicles of different sizes. The size and colour of surrounding vehicles denotes the level of risk. Hence, vehicles that are in the driver's blind spot are considered high risk and are represented by big red icons. Low risk cars are depicted with small green icons. High proximity or hidden vehicles at intersections are also high risk and hence are shown as big and red. Surrounding vehicles' positions and speeds can be obtained from on-board vehicle sensors. Vehicles at intersections can be obtained through vehicle-to-vehicle communication protocol. The prototype visualization metaphor presented in Figure 2 is depicted on the vehicles windshield. The second design (arrows) of the system is based on the

need to prioritize information based on risk level and aims to warn drivers of vehicles that are expected to emerge from side roads and are not yet visible or vehicles that are in driver's blind spot. This, as illustrated in Figure 2, is expressed using arrows, on the augmented reality windscreen, pointing to the direction of the imminent threat, and is depicted on the vehicle's smart windshield. The most critical threat is depicted on the screen so as not to split the attention of the drivers among competing risks. This gives extra time to drivers to react to critical situations

The assessment of SA is achieved through an experiment with subjects using the developed driving simulator and virtual SDAT in a 3D CAVE facility. During this stage, 17 participants were involved, each spending on average of 90 min to complete the experiment in the VR CAVE lab. The analysis of the data collected from the experiments aimed to assess the SA NFR for the two candidate SDAT designs.

Data was collected in three phases: before, during and after the experiment. During the pre-experimental phase, the Manchester Driving Style questionnaire (Reason et al., 1990) was used to elicit the driving style of participants along with their demographic information. At the post experimental phase, data collection focused on the evaluation of the two candidate designs using a series of questions on four constructs: functionality, information visualization, usability and usefulness. During the experiment, participants' SA was measured while they were driving in a pre-specified route in the artificial road network within the 3D driving simulator (Figure 1), both with and without the SDATs. In particular, participants were asked to consult the HUD SDAT as during the driving simulation surrounding vehicles engaged the host vehicle by either pulling in or stopping in front of the driver. During the drivers' engagement with the experimental conditions, phenotype behavioural data related to driver workload and SA was recorded. Driver related data was recorded in a log-file on a simulation time-step basis. Specifically, manifestations of workload, such as lateral deviations (Montella et al., 2011), attention level through an electroencephalography (EEG) measurement, lane change, headway, speed, acceleration, deceleration, breaking patterns and steering wheel angle, were recorded on a time-location log-file. Collected data was automatically assigned to road sections that were specified in advance by the analysts, based on infrastructural properties. The assessment of the drivers' SA was achieved using the SAGAT (Situation awareness global assessment technique) method, which uses objective measures of SA gathered during an interruption in task performance. Hence, during each scenario with the participant, the simulation was stopped (freeze) three times, at points on the road network where the three dangerous scenarios were unfolding (car pulling in from the left, car stopping in front, car pulling in from the right). At each simulation freeze, participants were asked to complete a questionnaire that inquired their understanding of the situation. During the freeze, the simulators screens were blank. The simulator saved several screenshot of the situation just before the freeze to enable the comparison between the 'actual' event and the subjects' perceived situation.

5 RESULTS

Data collected from the simulations were pre-processed and analysed to identify differences between the actual situation and the participants' perceptions of the situation under the three conditions and the three interventions (phases). Analysis was conducted on both the post-experiment and the experiment data. Results from the post experiment data revealed that both SA enhancement systems were perceived by the users as improvements over the control condition (i.e. without any enhancement). Specifically, the post-experiment questionnaire addressed the following dimensions of each candidate design: features, user interface, ease of learning, system capabilities, usefulness, ease of use, and SA. Each dimension was supported on average by 5 questions, on a 7 point response scale from 1 (negative effect) to 7 (positive effect). To increase the discrimination in the evaluators' judgment, participants' were asked to report the reasons for their choices and any interaction problems they had experienced under the relevant heuristic. Figure 3 shows the percentages of positive responses (i.e. >4, or <4 for negatively worded statements) for each of the measured dimension on which the two designs were evaluated. Based on this analysis, there do not seem to be noticeable differences in regards to user interface and ease of use. However, overall the radar design seems to have been perceived more positively than the arrows, especially in relation to learning, system capabilities, and usefulness. This might be attributed to the small size of the arrows that were popping up on the smart windshield. Among the two designs the radar design was also considered more appropriate to support driver SA. Moreover, based on open responses from participants, in certain occasions, the number of arrows that were present on the windshield were more than two. Hence, the cues were becoming destructing

rather than informative. On the other hand, the Radar design also had its shortcoming in terms of visualization of the threats. Specifically, the colouring and size of threats were considered insufficient.

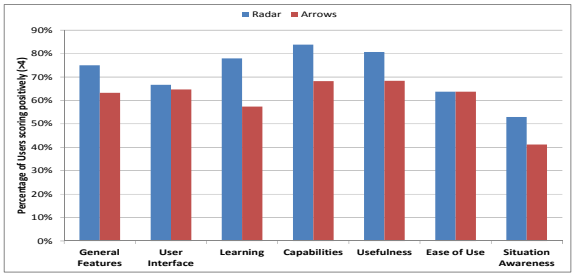


Figure 3: Percentages of positive responses in each of the measured dimensions, by design.

Results from data during the experiment aimed to assess the SA NFR using a combination of the SAGAT data and the driver behaviour data from the simulation log files. Initially the SAGAT and the driver behaviour data were integrated into one dataset for each participant. Subsequently the data that represented the actual situation was compared with the data that represented the perceived situation for each participant at each phase of the experiment. The similarity assessment between actual and perceived was estimated using the Euclidian distance metric. Analysis of the SA data was then performed using ANOVA in a within-subjects model. Based on the results, the use of both SDAT designs in an augmented reality overhead display demonstrated a superior performance to no-design. Results from the SAGAT analysis also revealed that design 1 (radar) was superior to design 2(arrows) and no design. This was identified as significant based on figure 4. In the same vein, the phase of the simulation freeze, denoting the sequence of the freeze, was also identified as a significant factor with phase 3 in the radar design having on average a SA metric of 85% compared to 63% in the control condition (no design).

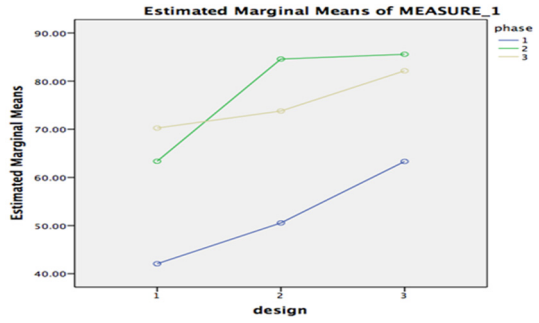


Figure 4: Estimated marginal mean for the 3 designs (radar -1, arrows-2, no-design-3) and the three phases of the simulation (freeze1-3).

6 CONCLUSIONS

The HF NFR validation method presented herein provides a novel cost effective solution to validating HF NFRs of prospective complex sociotechnical systems. It enables the evaluation of NFRs through experimentation in VR settings under an envelope of test scenarios. The developed driving simulator is component-based and hence enables the requirements engineer to easily customize it to the problem in hand. Requirements are realised in virtual settings and this provides designers with the flexibility of customizing the functionality of the SDAT in an attempt to satisfy the HF NFR under consideration. Results from the application of this method in the validation of the SA NFR of an in-vehicle SDAT revealed the method's practicality. The method is based on design science and encourages the redesign of the artefact until it satisfies the NFR. Results indicate that what the users experience during their interaction with the artefact and what they perceived of this experience as reported in the post-test questionnaire point to the same conclusion. Specifically, statistical analysis of the data collected indicated that the radar design is superior to arrows and no design. Similarly, subjective evaluation of the candidate designs also revealed the same results. Hence, this agreement is a good indication that the NFR validation method is producing accurate estimations. Limitations of this work concentrate on the simulator's level of realism and immersion factors that laboratory methods suffer from. Simulated settings do not currently offer the resolution of the real world, and so they may affect driving behavior. Future work will include the improvement of the realism factor which in turn will improve observational accuracy. Moreover, the experimental design for the evaluation of the SA was very time consuming. This could be optimized though the use of a cut down version of the SAGAT questionnaire.

ACKNOWLEDGEMENTS

This work was funded by the Cyprus Research Promotion Foundation, grand NEKYP/0311/02

REFERENCES

Adams K., et al, 2015, Non-functional Requirements in Systems Analysis and Design.in Topics in Safety, Risk, reliability and Quality, *Springer*.

Chung, L., Nixon, B. A., Yu, E. and J. Mylopoulos, 2000. Non-functional requirement in *Software Engineering, Kluwer Academic Publishing*.

Davenne D., Lericollais R., Sagaspe P., Taillard J., Gauthier A., Espié S., Philip P. 2012. *Reliability of simulator driving tool for evaluation of sleepiness, fatigue and driving performance, Accident Analysis and Prevention, 45*, pp.677-682.

Dingus, T. A., Klauer, S. G., Neale, V. L., Petersen, A., Lee, S. E., Sudweeks, J., et al. 2006. The 100-car naturalistic driving study: Phase II – Results of the 100-car field experiment. *Washington, DC: National Highway Traffic Safety Administration*.

Endlsey M. R 2012. *Designing for Situation Awareness: An Approach to User-Centered Design, Second Edition, CRS press.*

Fuller R. and Santos J. 2002. Human Factors for Highway Engineers. New York: Pergamon.

Gregoriades. A and Sutcliffe. A. 2007. Workload prediction for improved design and reliability of complex systems. Reliability. *Eng. System Safety, 39, n.4*, pp.530–549.

Gregoriades A, Sutcliffe A, Papageorgiou G, Louvieris P. 2010 Human-Centred Safety Analysis of Prospective Road Designs, *IEEE Transactions on Systems, Man and Cybernetics, Part A, Vol 40, 2*, pp 236-250.

Gregoriades A., Sutcliffe A. S., 2005. Scenario-based assessment of non-functional requirements, *IEEE Transactions on Software Engineering, , Vol 31, 5*, pp 392-409.

Hevner, A. & Chatterjee, S 2010. Design Research in Information Systems, Integrated Series in Information Systems, *vol 22, Springer*.

Holohan, C., Culler, R., & Wilcox, B. 1978. Effects of visual distraction on reaction time in a simulated traffic environment, *Human Factors, 20*, pp.409–413.

Illa, X. B., Franch, X, & Pastor, J.A. 2000. Formalising ERP selection criteria. In *Proceedings of the 10th international workshop on software specification and design (IWSSD'00)*, California, pp. 115–123.

Konstantopoulos P., Chapman P., Crundall D. 2010. Driver's visual attention as a function of driving experience and visibility. Using a driving simulator to explore drivers' eye movements in day, night and rain driving, Accident Analysis and Prevention, 42, pp.827-834.

Marew, T. et al. 2009. *Tactics based approach for integrating non-functional requirements in object-oriented analysis and design Syst. Software, 82*, pp. 1642–1656.

Matoussi, A., and Laleau, R. 2008. A Survey of Non-Functional Requirements in *Software Development Process, Report No. TR-LACL-2008-7, Departement d'Informatique Universite Paris 12, 2008*.

Montella A., Ariab M., D'Ambrosiob A., Galantea F., Maurielloa F., Pernettic M. 2011. Simulator evaluation of drivers' speed, deceleration and lateral position at rural intersections in relation to different perceptual cues, Accident Analysis and Prevention, 43, pp.2072-2084.

Nwokeji J., et al, 2014. ER 2014- International Workshp on Conceptual Modeling in Requirements and Business Analysis (MREBA), USA.

Peng, Y. G. Wang, H. Wang, 2012. User preferences based software defect detection algorithms selection using MCDM, *Information Sciences 191*. 3–13.

Reason, J., A. Manstead, S. Stradling, J. Baxter, K. Campbell. 2005. Errors and violations on the roads: a real distinction, *Ergonomics, 33 (10–11) (1990)*, pp. 1315–1332.

Stone, R. J. 2001. Virtual Reality for Interactive Training: An Industrial Practitioners Viewpoint, *International Journal of Human-Computer Studies, vol. 55*, pp. 699-711.

Sadana V., Liu X., 2007. Analysis of conflicts among non-functional requirements using integrated analysis of functional and non-functional requirements, *Computer Software and Applications Conference, COMPSAC*.

Zayaraz G, Thambidurai P., Srinivasan M., Rodrigues P.. 2005. *Software quality assurance through COSMIC FFP ACM SIGSOFT Software Engineering Notes, 30 (5).*

Marew, T. et al.2009. Tactics based approach for integrating on-functional requirements in object-oriented analysis and design Syst. Software, 82, pp. 1642–1656.

Zhu M., Luo X., Chen X., Dash W.,2012. A non-functional requirements tradeoff model in Trustworthy Software, *Information Sciences, Volume 191, 15*, pp 61-75.

Sutcliffe, A. and Gault, B. 2004, The ISRE method for analyzing system requirements with virtual prototypes. *Syst. Engin., 7*: 123–143.

Theeuwes J. et al, 2012, Designing Safe Road Systems: A Human Factors Perspective, *Ashgate*.

Estimating Trust in Virtual Teams
A Framework based on Sentiment Analysis

Guilherme A. Maldonado da Cruz, Elisa Hatsue Moriya Huzita and Valéria D. Feltrim
Informatics Department, State University of Maringá, Maringá, Paraná, Brazil
guilherme.maldonado.cruz@gmail.com, {emhuzita, vfeltrim}@din.uem.br

Keywords: Trust, Versioning System, Sentiment Analysis, Virtual Teams, Global Software Development.

Abstract: The advance in technology has enabled the emergence of virtual teams. In these teams, people are in different places and possibly over different time zones, making use of computer mediated communication. At the same time distribution brings benefits, there are some challenges as the difficulty to develop trust, which is essential for efficiency in these teams. In this scenario, trust information could be used to allocate members in a new team and/or, to monitor them during the project execution. In this paper we present an automatic framework for detecting trust between members of global software development teams using sentiment analysis from comments and profile data available in versioning systems. Besides the framework description, we also present its implementation for the GitHub versioning system.

1 INTRODUCTION

Software development using virtual teams characterizes distributed software development or global software development (GSD) when the distance between members comprises continents. It aims at providing benefits such as: low costs, proximity to the market, innovation and, access to skilled labor (O'Conchuir et al., 2006). However, geographic distribution and cultural differences bring some challenges as well, mainly in communication, which depends mostly on computer mediated communication (CMC).

One of the challenges faced by virtual teams and therefore GSD is the trust among team members. There are several studies that show the importance of trust for GSD teams (Kuo and Yu, 2009; Al-Ani et al., 2011; Pangil and Chan, 2014). Trust is related to the efficiency of the team, since high-trust teams can achieve their goals with less effort than low-trust teams. So, in this context, information about trust among people can be used for team recommendation and/or to monitor the relationship among members.

Some models have been proposed to estimate trust among people based on trust evidences, such as number of interactions, success of these interactions and similarity among people (Skopik et al., 2009; Li et al., 2010). We consider trust evidence something that indicates the existence of trust or that happens when there is trust among people.

Information used by trust models can be extracted, for example, from social networks. In general, it refers to the amount of interactions, evaluation of these interactions and their success. However, in a working environment people may not feel free to provide assessments of co-workers. Besides that, when the number of interactions is high, people may start to provide incorrect ratings, leading to incorrect trust estimation. Skopik et al. (2009) developed a set of metrics to analyze the success of an interaction. These metrics eliminates the need for feedback, however, they are domain dependent and ignore subjectivity, which is one of the characteristics of trust.

In this paper we present a framework to estimate trust among members of GSD teams. It extracts trust evidences observed in member interactions inside versioning systems, without human intervention and using sentiment analysis.

Sections 2, 3 and 4 present the concepts of GSD, trust and sentiment analysis, used in the development of the proposed framework. Section 5 describes the framework and Section 6 presents conclusions and also directions for future works.

2 GLOBAL SOFTWARE DEVELOPMENT

According to O'Conchuir et al. (2006) GSD is a collaborative activity, which can be characterized by

Cruz, G., Huzita, E. and Feltrim, V.
Estimating Trust in Virtual Teams - A Framework based on Sentiment Analysis.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 464-471
ISBN: 978-989-758-187-8

having members from different cultures and organizations, separated by time and space using CMC to collaborate. This team organization aims at providing benefits, such as: reduced development costs, follow-the-sun development, modularization of labor, access to skilled labor, innovation, best practices and knowledge sharing and proximity to the market.

Despite its benefits, GSD also brings challenges that add to those already existing in virtual teams, such as: strategic problems, cultural problems, inadequate communication, knowledge management, processes management and technical problems. Among these challenges is trust (Khan, 2012).

In this context, trust is important, mainly in GSD, due to members' inability to check what other members are doing by just watching (Jarvenpaa et al., 1998). Thus, trust reduces the risk and cost of monitoring (Striukova and Rayna, 2008). It also impacts in information sharing (Pangil and Chan, 2014), cohesion (Kuo and Yu, 2009) and cooperation (Striukova and Rayna, 2008).

3 TRUST

Trust has been studied in many fields, such as psychology, philosophy and economics. Based on definitions of different areas, Rusman et al. (2010, p.836) defined trust as:

> a positive psychological state (cognitive and emotional) of a trustor (person who can trust/distrust) towards a trustee (person who can be trusted/distrusted) comprising of trustors positive expectations of the intentions and future behavior of the trustee, leading to a willingness to display trusting behavior in a specific context.

This definition presents one of the trust properties, which is context specificity. Trust is also dynamic, non-transitive, propagative, composable, subjective, asymmetrical, events sensitive and self-reinforcing (Sherchan et al., 2013).

Before deciding to trust, a person evaluates the trustworthiness of the person to be trusted and the risk involved, so that if she chooses to trust, she became vulnerable positively and negatively to the trusted person, assuming the risk (Rusman et al., 2010). Therefore, the higher the trustworthiness, the higher the chance to be trusted. Rusman et al. (2010) define trustworthiness antecedents as attributes used to evaluate trustworthiness and divided them into five categories: (i) communality, (ii) ability, (iii) benevolence, (iv) internalized norms and (v) accountability.

3.1 Trust Evidence

Through a literature review we could not find an exact formula to determine whether there is trust among team members. However, some studies indicate behaviours and characteristics that serves as evidence of trust existence. For example, Jarvenpaa et al. (1998), conducted a qualitative study based on observations of teams with a high level of trust and teams with low level of trust. The authors observed common characteristics to high-trust teams that did not appear in low-trust teams, enumerated as: proactivity, task oriented communication, positive tone, rotating leadership, task goal clarity, roles division, time management, feedback and intensive communication.

Besides Jarvenpaa et al.'s (1998) work, we found other studies identifing teams characteristics which serve as evidence of trust. The list below sums up the trust evidences found in our literature review:

- Initiation and response: initiations are defined as questions or statements that lead the receiver to provide a relevant response. Iacono and Weisband (1997) used this characteristic to measure trust.
- Motivation: According to (Paul and He, 2012) motivation and trust are highly correlated.
- Knowledge sharing: Paul and He (2012) showed that the greater the trust among people, the greater is information sharing between them.
- Knowledge acceptance: people tend to accept knowledge of who they trust (Al-Ani et al., 2011).
- Trustworthiness: trust and trustworthiness are highly correlated (Jarvenpaa et al., 1998).
- Proactivity: high-trust team members are proactive, volunteering for roles and showing initiative (Jarvenpaa et al., 1998).
- Task oriented communication: in high-trust teams most conversations are about tasks to be completed (Jarvenpaa et al., 1998).
- Positive tone: high-trust teams tend to show enthusiasm in their conversations, praising and encouraging each other (Jarvenpaa et al., 1998).
- Task goal clarity: high-trust teams tend to discuss their goals, and when in doubt, they seek coordinators for clarification instead of making assumptions (Jarvenpaa et al., 1998).
- Rotating leadership: many members show leadership traits, and according to project needs, they assume the leadership as necessary (Jarvenpaa et al., 1998).
- Role division: team members assume roles in their project and show results of their work so others can provide feedback (Jarvenpaa et al., 1998).

- Time management: high-trust teams discuss deadlines, establish milestones and care to fulfill them (Jarvenpaa et al., 1998).
- Feedback and intense communication: high-trust teams display intense communication and feedback about team members' work (Jarvenpaa et al., 1998; Rusman et al., 2010; Kuo and Yu, 2009).
- High Performance: trust is positively related with team cohesion (Kuo and Yu, 2009), commitment, satisfaction and performance (Mitchell and Zigurs, 2009).
- Output quality: trust is positively related with output quality (Khan, 2012).
- Common vocabulary: when there is trust between people they tend to share a common vocabulary in CMC (Scissors et al., 2008).

Besides the evidences described above, Khan (2012) considered authority delegation, enthusiasm and high quantum of work as signs of trust. To Rusman et al. (2010), resources sharing, task division and delegation also occur when there is trust among team members.

3.2 Trust Models

In our literature review we found two models to estimate trust among people. The framework proposed by Skopik et al. (2009) aims at determining trust automatically, without the need for explicit feedbacks. The framework generates a graph in which nodes represent both services and people, and edges represent the trust value between them. Trust values are derived from the number of successful interactions relative to the total number interactions. Successful interactions are computed by a set of metrics, such as occurrence of errors in services.

The downside of Skopik et al.'s (2009) work is that, by relying on metrics, it ignores subjectivity that is intrinsic to trust by treating all people equally. For instance, if a service takes up to 30 seconds to respond an interaction, one person may consider it a success, while some other person may consider it a failure, even if it spends 10 seconds. Thus, this type of metric fails to capture such subjectivity.

The trust model proposed by Li et al. (2010) aims at assisting users of E-commerce in choosing best sellers. In their work, trust is estimated based on assessments made by users after interactions, and in the absence of interactions, on the similarity between assessments provided by them. It generates a user graph in which edges and weights represent the trust among them.

4 SENTIMENT ANALYSIS

Sentiment analysis have gained a lot of attention by the research community in the last decade, and have found its application in almost every business and social domain (Liu, 2012).

One of the tasks focused by sentiment analysis systems is to determine for a given text the polarity of the sentiment expressed: if it is positive, neutral or negative. The text unity used in the analysis determines its level, which usually falls into one of the three: (i) document level, (ii) sentence level and (iii) entity-aspect level (Liu, 2012).

Sentiment analysis has been also applied in the context of software development research. Guzman (2013) used sentiment analysis to capture emotion during diferent software development phases and to provide emotional climate awareness. Borbora et al. (2013) considered the sentiment expressed in communications as an indicator of the presence/absence of trust among stakeholders. Zhang et al. (2009) suggested the use of sentiment analysis to get a better understanding of trust among users. In fact, as presented in Section 3.1, one of the trust evidences is positive tone in communication, which can be directly captured by sentiment analysis tools.

5 THE FRAMEWORK

As previously discussed, virtual teams need trust among members in order to achieve their goals, since trust affects team's efficiency (Pangil and Chan, 2014). Trust models can be used to monitor trust among team members. However, some models require users to provide evaluation of others, or assume that there are means of informing if interactions were positive or not. The problem is that, in GSD teams, members can not feel free to evaluate co-workers. Even if it were not an issue, there would be a lot of interactions and members could end up getting tired of evaluating each interaction, providing nonsense evaluations that compromise the outcome of the models (Li et al., 2010).

In this context, we propose a framework to automatically estimate trust among GSD team members. The framework collects trust evidences observed in member interactions inside versioning systems, without human intervention and using sentiment analysis. Figure 1 presents the framework in terms of its inputs, used techniques, trust evidences considered and its output. The remaining of this section describes the framework focusing on its characteristics, how it works, its and design and implementation.

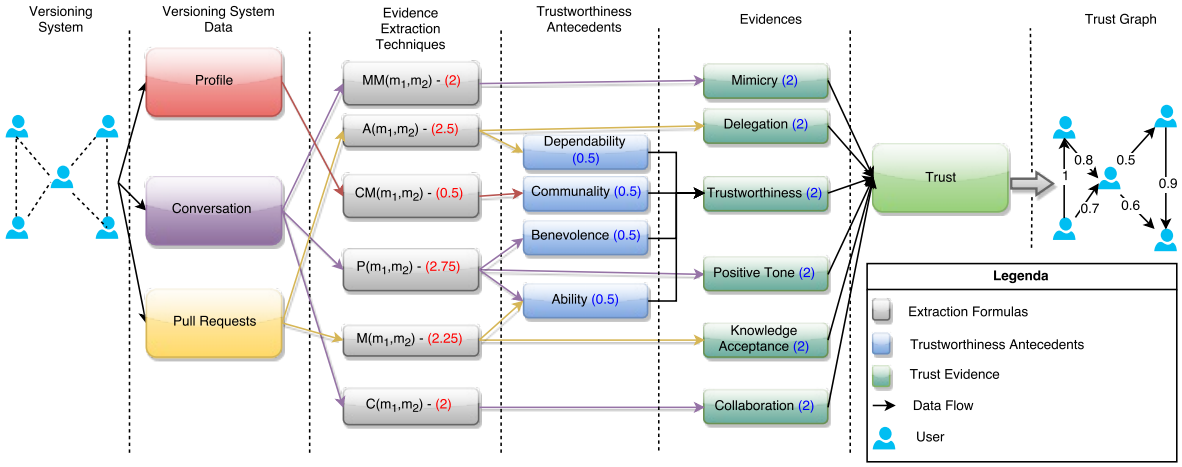


Figure 1: Proposed framework to estimate trust existence.

5.1 Characteristics

The main characteristics of the framework are:

a) It uses versioning systems as data source. Versioning systems are tools heavily used in software development and therefore in GSD (Robbes and Lanza, 2005). Particularly, the versioning systems that interest us are the ones that allow their users to perform commits and comment other's commits.

b) It uses trust evidences extracted from versioning system data (comments, commit state and user profile) to estimate trust among members of a project. As we could not extract all the trust evidences described in Section 3.1, the ones considered in the framework are: mimicry (common vocabulary), delegation, trustworthiness, positive tone, knowledge acceptance and collaboration. To extract the trustworthiness of a member we estimate four of the five trustworthiness antecedents presented in Rusman et al. (2010): dependability, communality, benevolence and ability.

c) It is automatic. Once it is set, the framework retrieves data without human intervention, estimates the existence of trust according to the temporal window and update estimated values according to the update frequency.

d) It preserves the subjectivity inherent to trust by using sentiment analysis and considering mimicry as a trust evidence. Sentiment analysis values and mimicry are extracted from how members write their comments, which is personal for each member. Thus, the framework takes into account subjectivity as it infers trust evidences from personal data. With sentiment analysis we can infer positive tone directly. Benevolence can be inferred since it was defined in Rusman et al. (2010) as the positive attitude and courtesy displayed by the trustee. Ability can also be inferred from sentiment analysis while the polarity of every comment may be seen as a feedback about the commit, and the commit in turn is the result of a member's ability to solve a problem. Therefore, we use the polarity of comments as a feedback about members' ability.

e) It updates evidences and trust values over time. It considers a time window to perform the extractions, and from time to time it moves this window, discarding old data, retrieving new ones and updating the values according to the data in the current time window.

f) It generates an initial graph of relations. This graph tells in which pull requests team members interact. The initial graph has an edge between a pair of members if they interact in a pull request. During execution, we add partial and final edges to the initial graph of relations. Partial edges keep partial values that are used to calculate final edges. One final edge is added for each evidence extraction technique. For instance, a pair of nodes may have many partial edges keeping the polarity of one comnent each, and one final edge keeping the rate of positive comments.

g) It generates a trust graph with its estimative of trust existence between each pair of members that interacted in at least one pull request. In this graph, nodes represent members of a project and edges represent the existence of trust between them. The weight of the edge, ranging from 0 to 1, displays the probability that there is trust between

two members. The closer to 1 the edge value is, the higher is the chance of existing trust between those two members.

Comparing our framework with the works presented in Section 3.2, we also use interactions like them both (Skopik et al., 2009; Li et al., 2010). Unlike Li et al. (2010), we removed the need for assessments, but kept the time factor by using a temporal window. We wanted it to be automatic like in Skopik et al. (2009), but we did not use metrics. Instead, we used mimicry and sentiment analysis in order to preserve subjectivity. We also added more trust evidences found in the literature in order to enrich the information used to estimate trust.

5.2 Design and Implementation

We designed and implemented an instance of our framework[1] to work with the GitHub versioning system. As presented in Figure 2, the framework is composed of four components:

`Graph` This component provides the framework with graphs that will be used as the initial graph of relations and the trust graph. By changing this component, we are able to alter how the framework keeps its data. The default implementation uses graphs.

`VS Data Extractor` This component extracts data from the versioning system. It extracts profile information from members in the project, pull requests information and conversations. In our implementation for GitHub we used the GitHub Java API[2]. Besides extracting data, this component also generates the initial graph of relations.

`Evidence Analyzer` This component provides classes implementing the *EvidenceAnalyser* interface representing an evidence extraction technique. Each one of these classes will analyze data extracted from versioning system and generate a value that is stored in the graph of relations and used to estimate trust existence. We provided six evidence extraction techniques: mimicry, assignments, communality, polarity, merges and collaboration. These evidence extraction techniques are formulas described further in this section. The addition of more evidences to the framework requires only a new implementation of *EvidenceAnalyser* interface to a new evidence extraction technique.

[1]Available at https://github.com/Tulivick/Trust-Framework

[2]https://github.com/eclipse/egit-github/tree/master/org.eclipse.egit.github.core

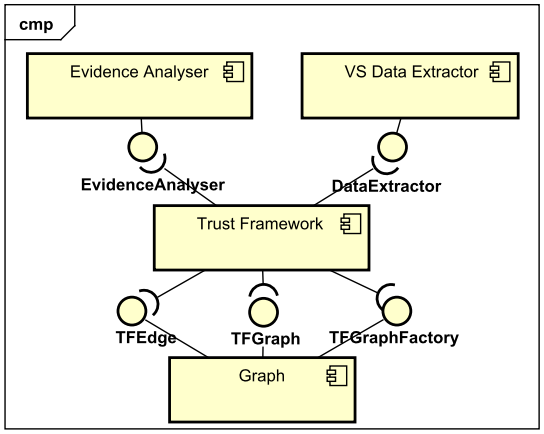


Figure 2: Component diagram for trust framework.

`Trust framework` This is the main component. It provides ways to configure and use the framework. This component is responsible to get data from `VS Data Extractor`, and transmit it to `Evidence Analyzer` with the graph of relations, so the graph can be updated. From the graph of relations, this component generates the trust graph using the formula described further in this section.

Note that by providing new implementations for `VS Data Extractor` we are able to extend the framework to other versioning systems. However, as each versioning system may have different data, the `Evidence Analyzer` is bound to the `VS Data Extractor` component. If one wants to provide support to another versioning system, it may be necessary to replace `Evidence Analyzer` by one that supports the new versioning system. It is also possible to extend the set of considered evidences by implementing other classes that extend *EvidenceAnalyser* interface.

As mentioned before, in order to extract evidence values, we used a set of formulas that we named evidence extraction techniques on Figure 1. To extract conversations mimicry, we calculate how similar are the vocabularies used in conversations for a pull request. We calculate the similarity of two comments vocabularies by using cosine similarity on word frequency. In Equation 1, $SC(c_1,c_2)$ is the cosine similarity between two comments, and FV is the words frequency array for each comment.

$$SC(c_1,c_2) = \frac{FV_1 \cdot FV_2}{\|FV_1\|\|FV_2\|} \quad (1)$$

The member m_1 mimicry value for the member m_2, $MM(m_1,m_2)$ is the average of the similarities between m_1's comments and m_2's comments that precede in the same pull request. In Equation 2 PR_{12} is the set of pull requests where m_1 and m_2 interact, C_1

is the comments set of m_1 in pull request pr_i and C_{2j} is the comments set of m_2 preceding the comment c_j.

$$MM(m_1,m_2) = \frac{\sum_{pr_i=1}^{|PR_{12}|}\sum_{c_j=1}^{|C_1|}\sum_{c_k=1}^{|C_{2j}|} SC(c_j,c_k)}{\sum_{pr_i=1}^{|PR_{12}|}\sum_{c_j=1}^{|C_1|}\sum_{c_k=1}^{|C_{2j}|} 1} \quad (2)$$

Communality is calculated by averaging the similarity of three GitHub user profile attributes: (I) followed users, (II) watched projects, and (III) location. (I) and (II) are calculated using Jaccard similarity (Equation 3) where C_1 and C_2 are evaluated sets, in this case the followed users and watched projects for both members. (III) in turn uses a variant of the Euclidean distance (Dodd et al., 2013, Equation 4) on Geert Hofsted index values (Hofstede et al., 2010). Indexes of Geert Hofsted characterize the culture of a region using six indexes: power distance, individualism, masculinity, uncertainty avoidance, long-term guidance and indulgence. In Equation 4 L_i is a location, K is the amount of indexes, I_{ik} is the value of the index k to a location L_i and V_k is the variance of the index k. If the indexes do not exist for a particular location, we will consider biggest the distance possible between two locations.

$$JS(C_1,C_2) = \frac{C_1 \cap C_2}{C_1 \cup C_2} \quad (3)$$

$$ED(L_1,L_2) = \sqrt{\sum_{k=1}^{K} \frac{(I_{1k}-I_{2k})^2}{V_k}} \quad (4)$$

Therefore, the communality $CM(m_1,m_{,2})$ between members m_1 and m_2 is given by Equation 5 where F_i, W_i and L_i are respectively the set of followed users, watched projects and m_i's location.

$$CM(m_1,m_2) = \frac{JS(F_1,F_2) + JS(W_1,W_2) + \frac{1}{1+ED(L_1,L_2)}}{3} \quad (5)$$

We used comments polarities to estimate benevolence, ability and positive tone. The polarity value between two members is given by Equation 6, where $P(m_1,m_2)$ is member m_1 polarity value for the member m_2, C_{+12} is the amount of comments with positive polarity from m_1 to m_2 and C_{12} is the total amount of comments from m_1 to m_2. Comments from m_1 to m_2 are comments that m_1 wrote in m_2's pull request or comments where m_1 mentioned m_2. Note that there are pull requests created by m_1 where m_2 did not interact, however these are not considered.

$$P(m_1,m_2) = \frac{C_{+12}}{C_{12}} \quad (6)$$

Dependability and delegation can be observed through the assignments of a member by another in a pull request. The assignment value of a member m_1 to a member m_2 is 1 if m_1 assigned at least one pull request to m_2 or 0 otherwise. Equation 7 calculates the assignment value for m_1 to m_2, $A(m_1,m_2)$, based on the amount of pull requests assigned to m_2 by m_1, PRs_{12}.

$$A(m_1,m_2) = \begin{cases} 1, & \text{if } PRs_{12} \geq 1 \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

We infer values for knowledge acceptance and ability of a member from the amount of pull requests that were merged. In Equation 8, $M(m_1,m_2)$ is the proportion of pull requests created by m_2 in which m_1 and m_2 interacted, that were merged, PR_{a12} is the amount of pull request created by m_2 in which m_2 and m_1 interacted, that were merged, PR_{12} is the amount of pull request created by m_2 in which m_1 and m_2 interacted.

$$M(m_1,m_2) = \frac{PR_{a12}}{PR_{12}} \quad (8)$$

We estimate collaboration as the proportion of interactions as given by Equation 9. In this equation $C(m_1,m_2)$ is the proportion of interactions between m_1 and m_2 out of the total interactions of m_1. I_{12} is the number of interactions between m_1 and m_2, and I_1 is the amount of m_1 interactions with everyone.

$$C(m_1,m_2) = \frac{I_{12}}{I_1} \quad (9)$$

Finally, we estimate values of trust among members using Equation 10, which is the weighted average of the formulas listed above. The best way to set the weights α_i would be through the use of history data from previous projects to learn the weights. However, we are not aware of any database with trust information among members in versioning systems. Thus we defined weights based on the number of evidences calculated through the use of each formula presented above.

$$T(m_1,m_2) = \frac{\alpha_1 MM(m_1,m_2) + \alpha_2 CM(m_1,m_2) + \alpha_3 P(m_1,m_2)}{\alpha_1+\alpha_2+\alpha_3+\alpha_4+\alpha_5+\alpha_6} + \frac{\alpha_4 A(m_1,m_2) + \alpha_5 M(m_1,m_2) + \alpha_6 C(m_1,m_2)}{\alpha_1+\alpha_2+\alpha_3+\alpha_4+\alpha_5+\alpha_6} \quad (10)$$

First we consider that every evidence has the same weight. We choose weight 2, because we will propagate it to the trustworthiness antecedents and evidence extraction techniques, and if it were lower we would obtain many decimal places.

We consider that each trustworthiness antecedent has the same weight to calculate trustworthiness, so by propagating the weight 2 from trustworthiness to its antecedents we obtain a weight of 0.5 for each.

By propagating the weights from antecedents and evidences we obtain the values $\alpha_1 = 2$, $\alpha_2 = 0.5$, $\alpha_3 = 2.75$, $\alpha_4 = 2.5$, $\alpha_5 = 2.25$ and $\alpha_6 = 2$ presented in front of each formula on Figure 1.

As an example, we will propagate the weights to $MM(m_1, m_2)$ and $A(m_1, m_2)$. $MM(m_1, m_2)$ infers only mimicry that has weight 2, thus when we propagate its weight to the formula, that gains weight 2 also. $A(m_1, m_2)$ in turn infers delegation with weight 2, and dependability with weight 0.5, thus by propagating this weights to the formula it gains weight 2.5. By doing this to each formula we obtained the α values.

In Figure 1, the numbers in parentheses represent the weights of each evidence and trustworthiness antecedents, except the ones appearing at the evidence extraction techniques, which are the α values obtained by propagating the weights from evidences and trustworthiness antecedents.

As mentioned at the beginning of this subsection, each formula is coded in a class implementing the `EvidenceAnalyser` interface in the `EvidenceAnalyser` component. In order to implement these classes, we used the following APIs: Lucene[3] to generate word frequency count for comments, Sentistrength[4] to retrieve the polarity for every comment, AlchemyApi[5] to extract comments targets and Google Maps Geocoding API[6] to discover from which country each user's address was.

5.3 How It Works

To start using the framework we need to configure it. This is done by informing a target project (owner/repository), the evidence extraction techniques to be used and their weights, the size of the temporal window, the update frequency, and a graph factory.

Once it starts running, the framework extracts project data from GitHub. These data are in turn processed using the formulas described in the previous section to extract the trust evidences. The data retrieved from GitHub are delivered to each instance of *EvidenceAnalyser* interface. This instances will add partial and final edges to the initial graph of relations. Combining final edges values through Equation 10 we estimate trust existence among members. This estimate is provided by means of a trust graph.

With the trust graph in hands, we can, for example, use trust values to suggest members to a team that has a higher chance of having a high level of trust. In addition, as the framework process the latest comments and automatically updates the values of trust, it enables us to monitor trust among members, so that the manager can take actions when negative changes in the teams' trust are perceived.

[3]https://lucene.apache.org/

[4]http://sentistrength.wlv.ac.uk/

[5]http://www.alchemyapi.com/

[6]https://developers.google.com/maps/

6 CONCLUSIONS

The efficiency of a GSD team is directly tied to trust among team members. The higher the trust is, the lower project costs are. Trust also increases communication and facilitate cooperation, coordination, knowledge and information sharing, which improve the quality of generated products.

Motivated by the importance of trust in these teams, this work presented an automatic framework to estimate trust existence among members of a GSD team. It uses versioning systems, a collaborative tool used in software development, as data source. To design the framework, we used trust evidences presented in the literature that can be extracted from versioning systems data. One of the main features of the proposed framework is the use of sentiment analysis to extract some of these evidences, for example, the positive tone of the conversations.

The main contribution of this paper is in the mapping of trust evidences and elements of trust models that can be captured using sentiment analysis. We also contributed with an implemented instance of the framework that works with GitHub. We expect our framework to provide a better estimative of trust existence than general automatic models in the literature since it uses sentiment analysis and a rich set of evidences. By employing sentiment analysis, we have added subjectivity to our estimative, which is an important characteristic of trust. GSD managers can benefit from our framework to create teams with higher trust levels. With our framework it is also possible to monitor trust level variations, so actions can be taken by the project manager when trust level decreases.

As we do not have a GitHub dataset annotated with trust information, we considered all weights the same. In order to better calibrate the weights we are conducting a survey with experienced people in GSD to aid us determine the weight of each evidence and validate the formulas we presented. The results obtained until now, are promising.

As future work we foresee: (i) the addition of other trust evidences to the framework, (ii) the conclusion and analyzes of the results for our survey, which may lead to an improvement of the framework and the conduction of a new survey, and (iii) the monitoring

of a real project, allowing us to collect trust information about team members in order to compare with the results given by our framework.

REFERENCES

Al-Ani, B., Wilensky, H., Redmiles, D., and Simmons, E. (2011). An understanding of the role of trust in knowledge seeking and acceptance practices in distributed development teams. In *6th IEEE International Conference on Global Software Engineering (ICGSE)*, pages 25–34.

Borbora, Z. H., Ahmad, M. A., Oh, J., Haigh, K. Z., Srivastava, J., and Wen, Z. (2013). Robust features of trust in social networks. *Social Network Analysis and Mining*, 3(4):981–999.

Dodd, O., Frijns, B., and Gilbert, A. (2013). On the role of cultural distance in the decision to cross-list. *European Financial Management*, pages n/a–n/a.

Guzman, E. (2013). Visualizing emotions in software development projects. In *First IEEE Working Conference on Software Visualization (VISSOFT)*, pages 1–4.

Hofstede, G., Hofstede, G. J., and Minkov, M. (2010). *Cultures and Organizations: Software of the Mind*. McGraw-Hill USA, New York, 3 edition. An optional note.

Iacono, C. and Weisband, S. (1997). Developing trust in virtual teams. In *Proceedings of the Thirtieth Hawaii International Conference on System Sciences, 1997*, volume 2, pages 412–420 vol.2.

Jarvenpaa, S. L., Knoll, K., and Leidner, D. E. (1998). Is anybody out there? antecedents of trust in global virtual teams. *Journal of Management Information Systems*, 14(4):pp. 29–64.

Khan, M. S. (2012). Role of trust and relationships in geographically distributed teams: exploratory study on development sector. *International Journal of Networking and Virtual Organisations*, 10(1):40–58.

Kuo, F.-y. and Yu, C.-p. (2009). An exploratory study of trust dynamics in work-oriented virtual teams. *Journal of Computer-Mediated Communication*, 14(4):823–854.

Li, J., Zheng, X., Wu, Y., and Chen, D. (2010). A computational trust model in c2c e-commerce environment. In *Proceedings - IEEE International Conference on E-Business Engineering, ICEBE*, pages 244 – 249, Shanghai, China.

Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 5(1):1–167.

Mitchell, A. and Zigurs, I. (2009). Trust in virtual teams: Solved or still a mystery? *ACM SIGMIS Database*, 40(3):61–83.

O'Conchuir, E., Holmstrom, H., Agerfalk, P., and Fitzgerald, B. (2006). Exploring the assumed benefits of global software development. In *International Conference on Global Software Engineering, 2006. ICGSE '06*, pages 159–168.

Pangil, F. and Chan, J. (2014). The mediating effect of knowledge sharing on the relationship between trust and virtual team effectiveness. *Journal of Knowledge Management*, 18(1):92–106.

Paul, S. and He, F. (2012). Time pressure, cultural diversity, psychological factors, and information sharing in short duration virtual teams. In *45th Hawaii International Conference on System Science (HICSS)*, pages 149–158.

Robbes, R. and Lanza, M. (2005). Versioning systems for evolution research. In *Eighth International Workshop on Principles of Software Evolution*, pages 155–164.

Rusman, E., van Bruggen, J., Sloep, P., and Koper, R. (2010). Fostering trust in virtual project teams: Towards a design framework grounded in a TrustWorthiness ANtecedents (TWAN) schema. *International Journal of Human-Computer Studies*, 68(11):834–850.

Scissors, L. E., Gill, A. J., and Gergle, D. (2008). Linguistic mimicry and trust in text-based cmc. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work*, CSCW '08, pages 277–280, New York, NY, USA. ACM.

Sherchan, W., Nepal, S., and Paris, C. (2013). A survey of trust in social networks. *ACM Computing Surveys (CSUR)*, 45(4):47:1–47:33.

Skopik, F., Truong, H. L., and Dustdar, S. (2009). Viete - enabling trust emergence in service-oriented collaborative environments. In *Proceedings of the Fifth International Conference on Web Information Systems and Technologies WEBIST*, pages 471–478.

Striukova, L. and Rayna, T. (2008). The role of social capital in virtual teams and organisations: corporate value creation. *International Journal of Networking and Virtual Organisations*, 5(1):103–119.

Zhang, Y., Chen, H.-J., Jiang, X.-H., Sheng, H., and Wu, Z.-H. (2009). RCCtrust: A combined trust model for electronic community. *Journal of Computer Science and Technology*, 24(5):883–892.

A Language for Defining and Detecting Interrelated Complex Changes on RDF(S) Knowledge Bases

Theodora Galani[1,2], George Papastefanatos[1] and Yannis Stavrakas[1]
[1]*Institute for the Management of Information Systems, RC ATHENA, Artemidos 6 & Epidavrou, Marousi, Greece*
[2]*School of Electrical and Computer Engineering, NTUA, Zografou, Athens, Greece*
theodora@dblab.ece.ntua.gr, {gpapas, yannis}@imis.athena-innovation.gr

Keywords: Change Management, Data Evolution, RDF(S).

Abstract: The dynamic nature of web data brings forward the need for maintaining data versions as well as identifying changes between them. In this paper, we deal with problems regarding understanding evolution, focusing on RDF(S) knowledge bases, as RDF is a de-facto standard for representing data on the web. We argue that revisiting past snapshots or the differences between them is not enough for understanding how and why data evolved. Instead, changes should be treated as first-class-citizens. In our view, this involves supporting semantically rich, user-defined changes that we call complex changes, as well as identifying the interrelations between them. In this paper, we present our perspective regarding complex changes, propose a declarative language for defining complex changes for RDF(S) knowledge bases, and show how this language is used to detect complex change instances among dataset versions.

1 INTRODUCTION

The increasing amount of information published on the web poses new challenges for data management. A central issue concerns evolution management. Data published on the web frequently change, as errors may need to be fixed or new knowledge has to be incorporated. Data consumers need to know what changed among versions, as well as how and why it changed. Thus, the need for maintaining data versions and identifying changes becomes evident.

In this paper we focus on interpreting evolution on RDF(S) knowledge-bases, as RDF is the de-facto standard for representing data on the web. A typical approach for handling changes among dataset versions is computing diffs between them, leading to a machine-readable representation of changes based on triple additions and deletions. This approach does not provide any intuition about change semantics or possible relations between them. An ideal approach would compute human-readable, semantically rich changes along with any interrelations between them.

For example, consider a simplified ontology representing a company's employees, as in Figure 1. Figure 1(a) depicts the initial version, while Figure 1(b) the version after the modifications. Note that classes are in bold font. Each employee is described by her name, salary, position and optionally grade and projects assigned. Employees are organised in a hierarchical structure, depicting position hierarchy, as each one refers to another. In Figure 1(b), modified parts are depicted in light grey. Initially, employee "theo" is leading a small team of programmers, comprising of "mary" and "kate" working on project A. Later, he gets an excellent appraisal turning his grade from 9 to 10. As a result, he gains a salary increase. Also, he gets promoted to a manager. The promotion leads to an additional salary increase and overall the salary doubles. As the business goes well, a new employee has to be hired in order to organize the increasing team responsibilities. As a result, a new team leader is added, "nick", serving as senior employee, guiding "mary" and "kate", and reporting to the manager. From now on, the projects are assigned to him and thus they are moved from "mary" and "kate" to him.

Computing the diff between these two versions totally misses capturing change semantics and dependencies. Understanding the intentions behind data modifications can be even more complicated for large datasets, where changes are numerous and dispersed. Instead, Figure 2 depicts an intuitive and descriptive representation of how data changed. Figure 2(a) depicts the modifications regarding "theo", while Figure 2(b) regarding "nick". Each

Galani, T., Papastefanatos, G. and Stavrakas, Y.
A Language for Defining and Detecting Interrelated Complex Changes on RDF(S) Knowledge Bases.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 472-481
ISBN: 978-989-758-187-8

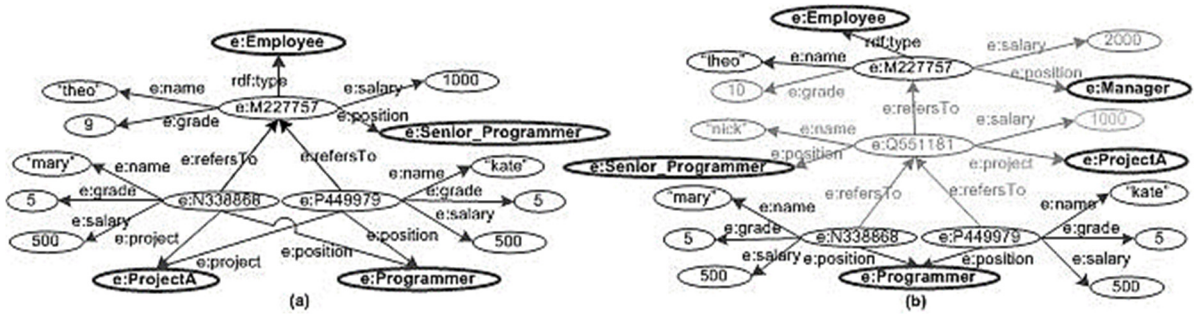


Figure 1: Simplified part of an employee ontology. (a) Initial version. (b) Version after modifications.

node represents a *change instance* detected between the aforementioned dataset versions. Change instances on leaf nodes (in grey) are fine-grained and model-specific, meaning that they do not comprise of other change instances and their semantics suit to the RDF data model. Each one corresponds to an added or deleted triple, having a suitable name and descriptive parameters. They are *simple change instances*. The rest change instances (in white) are coarse-grained and application/data-specific, meaning that they demonstrate structure and semantics suitable to the employee example. The hierarchical structure indicates that a change instance is build on top of others, demonstrating relations and dependencies among changes. They are *complex change instances*.

Consider the change instances Add_Grade and Delete_Grade in Figure 2(a). They serve as specializations of the model specific Add_Property _Instance and Delete_Property_Instance, respectively. This holds for all similar change instances regarding employee properties. Employee_Positive_Appraisal instance *contains* them, modelling the positive evaluation that took place. Similarly, Employee_Promotion_Manager and Employee_Salary_Increase group change instances providing richer semantics on how data changed. Employee_Salary_Increase is caused by Employee_Positive_Appraisal and Employee_Promotion_Manager. Causality is modelled on top of these changes through Salary_Increase_after_Positive_Appraisal and Salary_Increase_after_Promotion_Manager. These change instances are *overlapping* as they share a common part, Employee_Salary_Increase, modelling that they cause the same effect on data. Similar properties are demonstrated on change instances of Figure 2(b). Add_Employee instance groups all properties related to a newly added employee. Add_Senior_Employee instance is a specialization of Add_ Employee, where the added employee (with id e:Q551181, i.e. "nick") reports to a manager (with id e:M227757, i.e. "theo") and serves as a leader to other employees (with ids e:N338868 and e:P449979, i.e. "mary" and "kate"). This is modelled by the position he gets in the hierarchy, via Add_Reference instances. Also, Add_Senior_Employee instance contains a Move_Project_Assignment instance, as project A is moved from "mary" and "kate" to "nick", and Delete_Reference instances as these employees initially had "theo" as a leader. These changes are secondary and may happen when adding a senior employee.

In this paper, we argue that for understanding data evolution, changes should be treated as first-class-citizens. In our view, this involves supporting human-readable, semantically rich, user-defined changes, named *complex changes*. These changes are application/data-specific and coarse-grained, defined over primitive and model-specific changes, named *simple changes*. Modelling explicitly complex changes provides additional information for interpreting past data, while supporting user-defined changes allows interpreting evolution in multiple ways. On top of this, supporting *interrelated* complex changes, through nesting and overlaps, is an additional feature that enriches the complex changes' expressivity. A complex change may be part of another, may generalize/ specialize another, may cause another or may provide supplementary interpretation of evolution. Section 3 contains the basic concepts of our approach. Given these concepts, we provide a declarative language for defining complex changes (Section 4). We then define a process for detecting complex change instances among dataset versions (Section 5). Both the language and detection algorithm are influenced by our main contribution of supporting interrelated complex changes. Section 2 discusses related work and Section 6 concludes the paper.

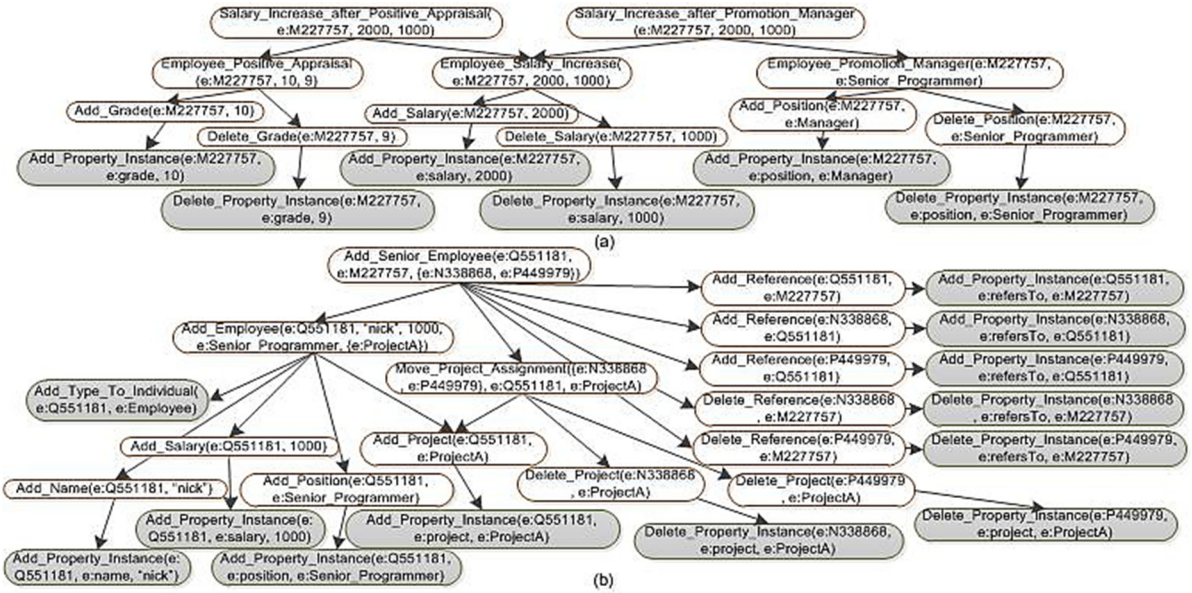


Figure 2: Hierarchy of detected simple and complex change instances (in grey and white fill respectively) of the employee ontology of Fig. 1. (a) Change instances regarding "theo". (b) Change instances regarding "nick".

2 RELATED WORK

A number of works focus on computing the differences between knowledge bases. In (Berners-Lee and Connolly, 2004) an ontology for representing differences between RDF graphs, in the form of insertions and deletions, is proposed. RDF graphs comparison is discussed, as well as updating a graph from a set of differences. In (Volkel et al., 2005) two diff algorithms are proposed: one computing a structural diff (as the set-based difference of the triples explicitly recorded into the two graphs), and one semantic diff (taking into consideration the semantically inferred triples). In (Franconi et al., 2010), an approach for computing a semantic diff is proposed, focusing on propositional logic knowledge bases but also being applicable to more expressive logics. A number of desired properties are discussed, like semantic diff uniqueness, the principal of minimal change, the ability to undo changes and version reconstruction. Similar properties are supported in (Zeginis et al. 2011), which focuses on computing deltas over RDF(S) knowledge bases. In (Noy and Musen, 2002; Klein, 2004), a fixed point algorithm for detecting ontology change is proposed. It employs heuristic-based matchers, introducing uncertainty to results.

Other works focus on supporting human-readable changes. In (Papavasileiou et al., 2013), a set of predefined high-level changes for RDF(S) knowledge bases and an algorithm for their detection are proposed. Changes verify the properties of completeness and unambiguity, for guaranteeing that every added or deleted triple is consumed by one detected high-level change and that detected high-level changes are not overlapping, respectively. In (Roussakis et al., 2015), an extension of (Papavasileiou et al., 2013) is proposed, providing a more generic change definition framework, based on SPARQL queries. In (Plessers, De Troyer and Casteleyn, 2007), Change Definition Language is proposed for defining and detecting changes over a version log using temporal queries. In (Auer and Herre, 2007) a framework for supporting evolution in RDF knowledge bases is discussed. Changes are triple additions and deletions and aggregated triples, resulting in a hierarchy of changes. However, neither a detection process, nor a specific language of changes is defined. In (Klein, 2004), an extension of (Noy and Musen, 2002) is proposed for detecting some of the proposed basic and composite changes. In general, (Klein, 2004), (Papavasileiou et al., 2013) and (Stojanovic, 2004) provide human readable changes in similar categories regarding granularity and semantics.

Our approach focuses on human readable changes. A visionary work was presented in (Galani

et al., 2015). Similar to (Klein, 2004), (Papavasileiou et al., 2013), (Roussakis et al., 2015) and (Stojanovic, 2004) we assume primitive changes, as simple changes, and groupings of them, as complex changes. Instead of providing a predefined list of complex changes, we support user-defined complex changes in order to capture richer semantics and multiple interpretations of evolution, as (Plessers, De Troyer and Casteleyn, 2007) and (Roussakis et al., 2015). Our main contribution is supporting interrelated complex changes providing a language for defining complex changes over simple or other complex changes and an appropriate detection algorithm. (Plessers, De Troyer and Casteleyn, 2007) and (Roussakis et al., 2015) do not support interrelated complex changes.

3 SIMPLE AND COMPLEX CHANGES

Modelling changes as first class citizens involves taking into account granularity and semantics of changes. Granularity poses the question of having fine-grained or coarse-grained changes. Fine-grained changes have the advantage of describing primitive changes, while coarse-grained changes provide semantics and conciseness by grouping primitive changes in logical units. Semantics poses the question of having model-specific or application/data-specific changes. Model-specific changes describe modifications that appear in a specific model, constituting a fixed set of generic changes. Application/data-specific changes suit specific use-cases and may be user-defined, allowing multiple interpretations of evolution.

As a result, we distinguish between simple and complex changes. Simple changes constitute a fixed set of fine-grained, model-specific changes. Complex changes are coarse-grained, user-defined, application/data-specific changes providing richer semantics on how data changed. Definitions 1 and 2 formally define simple and complex changes.

Definition 1: A simple change s is a tuple (n, p), where:

- n is the name of s, which must be unique.
- p is the list of descriptive parameters of s, where each one has a unique name within s.

Definition 2: A complex change c is a quadruple (n, p, D, F), where:

- n is the name of c, which must be unique and different from the simple change names.
- p is the list of descriptive parameters of c, where each one has a unique name within c.
- D is the set of simple (D_S) and complex changes (D_C) that c comprises of, where

$D = D_C \cup D_S$, $D_C \cap D_S = \emptyset$ and $D \neq \emptyset$.

- F is the set of constraints (F_C) that changes in D verify and bindings (F_B) specifying the parameters in p, where $F = F_C \cup F_B$ and $F_C \cap F_B = \emptyset$. Constraints are on changes (F_C^{car}) or change parameters (F_C^{par}), where $F_C = F_C^{car} \cup F_C^{par}$ and $F_C^{car} \cap F_C^{par} = \emptyset$.

For simple changes we rely on (Papavasileiou et al., 2013). Appendix summarizes the simple changes considered. They verify completeness and unambiguity properties, constituting a first layer of human-readable changes. Simple changes are additions, deletions and terminological changes (rename, split, merge) of RDF(S) entities (classes, properties, individuals). As stated, simple changes are fine-grained, i.e. they cannot be decomposed in more granular changes. This holds for additions/ deletions, but not for terminological changes, as they can be expressed as additions/ deletions plus extra conditions. For example, a class rename can be considered as an add class plus a delete class, which have the same "neighbourhood" (properties, connections to classes). However, we prefer them as simple changes in order to distinguish at simple change level real additions/deletions from virtual ones representing terminological changes. Thus, simple changes' set is not minimal.

A complex change is defined in terms of simple or other complex changes verifying constraints. Constraints specialize its meaning and are divided into those defined on changes and those on change parameters. Bindings specify complex change parameter values. Section 4 includes more details.

The ultimate goal of supporting simple and complex changes is detecting actual instances between dataset versions. Detection process leads into instantiating change parameters with values, indicating that specific data elements have been affected by a change in a specific manner. Definitions 3 and 4 define simple and complex change instances. Figure 2 presents simple and complex change instance examples.

Definition 3: A simple change instance of a simple change (n, p), is a tuple (n, v) where v is an instantiation of the parameters p.

Definition 4: A complex change instance of a complex change (n, p, D, F), is a tuple (n, v) where v is an instantiation of the parameters p.

For simple change detection we rely on (Papavasileiou et al., 2013). For complex changes we provide an algorithm in Section 5. Definition 5 defines when a complex change instance is detected. Definitions 6 and 7 define possible relations among change instances, as interrelations between changes are reflected on them.

Definition 5: Let $c = (n, p, D, F)$ be a complex change and V_b and V_a two dataset versions. A complex change instance $c_i = (n, v)$ is detected if for all changes in D instances are detected between V_b and V_a, forming D_i, such that constraints in F_C are verified on D_i, V_b and V_a, bindings in F_B applied on D_i form v and D_i is maximal.

We say that the set of change instances D_i corresponding to c_i verifies the complex change c.

Definition 6: Let c_i be an instance of complex change c and D_i the corresponding set of change instances verifying c. c_i contains the change instances in D_i.

Definition 7: Let c_i and c_i' be two complex change instances, where c_i does not contain c_i' and vice versa. They are overlapping if they both contain at least one common simple or complex change instance.

Containment is transitive. Complex change instances may form a hierarchy due to containment and overlaps, as in Figure 2.

4 A LANGUAGE FOR DEFINING COMPLEX CHANGES

We believe that an intuitive, user-friendly language based on change semantics should be provided for defining complex changes. Complex change definitions are then used for detecting respective instances. In this section, we propose a declarative language for defining complex changes. We provide its syntax by means of EBNF specification (Table 1) and some illustrative examples (Table 2) concerning the employee ontology in Figure 1.

A complex change definition is composed by a heading and a body. The heading contains a unique name and a list of descriptive parameters. The body contains a list of changes that the complex change comprises of, constraints on the changes appearing in the list and their parameters, and parameter bindings declaring how complex change parameters are evaluated. Constraints and bindings are optional. A complex change definition is nested if complex changes appear in its change list. Thus, complex changes are defined as interrelated. Constraints are divided into cardinality, testing value, relational, pre/post-conditions and functions.

Cardinality constraint determines whether zero, one or multiple instances of a specific change are to be grouped into a complex change. In case of one or

multiple change instances, the change is defined as mandatory. In case of zero instances the change is defined as optional, and if no instance is detected, the respective complex change can be still detected. Thus, complex changes are flexible and tolerant in partially performed modifications of minor significance. Posing a cardinality constraint is optional. If it is not defined, the default case is one change instance for the respective change. The following notations hold: at least one change instance "+", zero or one "?", zero or more "*".

Parameter bindings determine how complex change parameters are evaluated. In general, a complex change parameter equals a parameter of a change in its change list. However, recall that due to cardinality constraints multiple change instances of a specific change type may be grouped. In such case, a complex change parameter equals the union of the parameter values for all the change instances of a specific type grouped. As a result, complex change parameters are distinguished into those that evaluate into type set and those that evaluate into scalar values. In order to distinguish the parameter types, parameters evaluating into scalar values start with a lowercase letter, while those evaluating into sets with an uppercase letter. Parameter bindings are optional, in case they can be inferred by repeating each parameter into the contained changes and respective constraints.

Testing value constraints, relational constraints, pre/post-conditions and functions are constraints defined on change parameters. Testing value constraints limit a parameter value against a given constant, while relational constraints involve two change parameters defining how changes are connected. For these constraints binary operators are supported. Pre/post-conditions define how parameters are related in the version before (V_b) or after (V_a) the change, stating whether a triple must or must not exist in the version before or after. If a triple may be inferred in a version, this is denoted by the flag "inferred". Constraints may also be in the form of predefined functions of return type boolean. For example consider common functions on strings, like contains, which checks whether a string contains another given string. Constraints may form composite conditions, when combined in boolean expressions using logical and, or, not.

As complex changes are used in nested definitions and complex change parameters may evaluate into set or scalar values, we support binary operators between sets and between sets and scalar values. Also, in order to write conditions on set elements we use quantified expressions, which may

Table 1: The EBNF specification of the complex change definition language.

```
complex-change-definition = 'CREATE COMPLEX CHANGE' heading '{' body '}'';' ;
heading = name '(' parameter-list ')' ;
parameter-list = identifier {', ' identifier} ;
body = change-list ['; ' filter-list] ['; ' binding-list] ;
name = STRING ;
identifier = LETTER {LETTER|DIGIT} ;
change-list = 'CHANGE LIST' change {', ' change} ;
change = change-heading [cardinality] ;
change-heading = change-name '(' parameter-list ')' ;
change-name = name | NAMES OF SUPPORTED SIMPLE CHANGES ;
cardinality = '+'|'?'|'*' ;
filter-list = 'FILTER LIST' or-filter-expr {', ' or-filter-expr} ;
or-filter-expr = and-filter-expr {'||' and-filter-expr} ;
and-filter-expr = neg-filter-expr {'&&' neg-filter-expr} ;
neg-filter-expr = ['!'] filter-expr ;
filter-expr = bracketed-expr | expr ;
bracketed-expr = '(' or-filter-expr ')' ;
expr = [quantification] constraint ;
quantification = 'for' ('each'|'some'|'none') identifier 'in' identifier ':' ['for'
('each'|'some'|'none') identifier 'in' identifier ':'] ;
constraint = test-val-constr | rel-constr | pre-post-constr | fun-constr ;
test-val-constr = identifier bin-op constant ;
rel-constr = identifier bin-op identifier ;
pre-post-constr = '(' (identifier | value) ', ' (identifier | value) ', ' (identifier |
value) ')' ['inferred'] ('in' | 'not in') ('Vb' | 'Va') ;
fun-constr = name '(' parameter-constant-list ')' ;
bin-op = '=' | '!=' | '>' | '<' | '>=' | '<=' | 'subSet' | 'properSubset' | 'superSet' |
'properSuperset' | 'in' | 'not in' ;
parameter-constant-list = (identifier | constant) {', ' (identifier | constant)} ;
constant = set | value ;
set = '{' value-list '}' ;
value-list = value {', ' value} ;
value = URI | LITERAL ;
binding-list = 'BINDING LIST' binding {', ' binding} ;
binding = binding-equality | binding-union ;
binding-equality = identifier '=' identifier ;
binding-union = identifier '<-' identifier ;
```

Table 2: Complex change definitions regarding the employee ontology of Figure 1.

```
CREATE COMPLEX CHANGE Add_Grade(x, g) {
CHANGE LIST Add_Property_Instance(x, prop, g);   FILTER LIST prop="e:grade";   };
CREATE COMPLEX CHANGE Employee_Positive_Appraisal(x, ng, og) {
CHANGE LIST Add_Grade(x, ng), Delete_Grade(x, og);   FILTER LIST ng>og;   };
CREATE COMPLEX CHANGE Employee_Promotion_Manager(x, op) {
CHANGE LIST Add_Position(x, np), Delete_Position(x, op);   FILTER LIST np=e:Manager;   };
CREATE COMPLEX CHANGE Employee_Salary_Increase(x, ns, os) {
CHANGE LIST Add_Salary(x, ns), Delete_Salary(x, os);   FILTER LIST ns>os;   };
CREATE COMPLEX CHANGE Salary_Increase_after_Positive_Appraisal(x, ns, os) {
CHANGE LIST Employee_Salary_Increase(x, ns, os), Employee_Positive_Appraisal(x, ng, og); };
CREATE COMPLEX CHANGE Salary_Increase_after_Promotion_Manager(x, ns, os) {
CHANGE LIST Employee_Salary_Increase(x, ns, os), Employee_Promotion_Manager(x, op); };
CREATE COMPLEX CHANGE Move_Project_Assignment(S, c, val) {
CHANGE LIST Add_Project(c, val), Delete_Project(s, val) +;   BINDING LIST S←s;   };
CREATE COMPLEX CHANGE Add_Employee(x, name, salary, position, grade, Project) {
CHANGE LIST Add_Type_To_Individual(x, t), Add_Name(x, name), Add_Salary(x, salary),
Add_Position(x, position), Add_Grade(x, grade)?, Add_Project(x, project)*;
FILTER LIST t=e:Employee;   BINDING LIST Project←project;   };
CREATE COMPLEX CHANGE Add_Senior_Employee(seniorx, m, X) {
CHANGE LIST Add_Employee(seniorx, name, salary, position, grade, Project),
Add_Reference(seniorx, m), Add_Reference(x, seniorx)+, Delete_Reference(x, psx)*,
Move_Project_Assignment(S, seniorx, val)*;   FILTER LIST for each s in S:
(s,e:refersTo,seniorx) in Va, (m,e:position,e:Manager) in Va;   BINDING LIST X←x;   };
```

be in the form $\{\forall,\exists,\nexists\}\, x \in X : f(x)$ or $\{\forall,\exists,\nexists\}\, x \in X : \{\forall,\exists,\nexists\}\, y \in Y : f(x,y)$, where $f(x)$ and $f(x,y)$ are constraints on parameters evaluating into scalar values.

Table 2 contains complex change definitions regarding the changes of the employee ontology in Figure 1 discussed in introduction. Add_Grade models the case where a new grade property with value g is assigned to employee x. The changes it comprises of are declared in the "change list", while constraints in the "filter list". Add_Grade is a specialization of simple change Add_Property_ Instance, where the property equals to "e:grade". This is a testing value constraint over parameter prop. Notice that no binding is defined explicitly, as they are inferred by repeating complex change parameters as parameters of the changes in change list. Besides Add_Property_Instance no cardinality

constraint is defined, meaning that cardinality one is inferred. Similar complex change definitions for all employees' properties can be given, but are omitted due to space limitations. Employee_ Positive_Appraisal models the case when an employee x gets a new grade, ng, greater than the old one, og. It comprises of Add_Grade, so that the new grade is assigned to the employee, and Delete _Grade, so that the old grade is removed, both referring to the same employee x. A relational filter compares the new and the old grade. Empployee_Salary_Increase is similarly defined. Employee_Promotion_Manager models the case when an employee x becomes a manager. Add_ Position assigns the new position to x and Delete_ Position deletes the old one. A testing value constraint specifies the new position as e:Manager.

The complex change Salary_Increase_after_ Positive_Appraisal comprises of Employee_Salary _Increase and Employee_Positive_Appraisal, modelling the case when a salary increase of employee x is caused after receiving positive appraisal. Thus, complex changes are grouped due to a causality relation. A similar concept holds for Salary_Increase_after_Promotion_Manager. These changes both base on Employee_Salary_Increase, as they try to explain why this increase has been caused. Thus, respective instances may overlap, if they both refer to the same employee, like "theo" in Figure 1 and 2. Due to nested definitions the respective instances lead to a hierarchical structure.

Move_Project_Assignment models the case where a project val, initially assigned to a set of employees S, is later assigned to another employee c. It comprises of Add_Project, as the project is assigned to c, and Delete_Project, as the project is deleted from another employee s. Both changes refer to the same project, as val is repeated in both. Besides Delete_Project "+" is noted. This is a cardinality constraint defining that there might be multiple deletions. The project may be initially assigned to multiple employees and then deleted from many of them. In such case, all these Delete_Project instances will be grouped into the respective complex change instance (through detection process). Now, consider that similarly the project can be moved to multiple employees too. This would cause multiple Add_Project instances. But, on Add_Project it is assumed cardinality one. Therefore, only one instance will be grouped in every complex change instance and multiple complex change instances will be detected, one for each Add_Project instance. As a result, supporting cardinality is important in order to define how change instances are grouped. We choose to follow cardinality as in Table 2 in order to construct groupings per project and per employee it has been moved to. Due to cardinality constraint, parameter S holds all employee' ids that the project has been removed from, as defined in the binding list.

Add_Employee models the case where a new employee is added with a number of descriptive properties. x is of type e:Employee, as defined in the testing value constraint. Property grade is optionally added, as defined by "?" besides Add_ Grade. Add_Project is optional too, but if it is added there might be many instances ("*"). Add_ Senior_Employee is a specialization of Add_ Employee and thus it is defined on top of it. It models the case when a newly added employee refers to a manager and leads other employees. This is described by e:refersTo property, through Add_Reference changes. The fact that the added employee refers to a manager is defined by the second post-condition. Also, it is likely that projects are moved to the added employee from the employees he leads. This is demonstrated by Move_Project_Assignment and the first post-condition. A quantified expression is used in order to write the post-condition on the elements of set S.

5 COMPLEX CHANGE DETECTION

Complex change detection is the process of identifying complex change instances. It requires as input a set of simple change instances detected between two dataset versions (S_i), the actual dataset versions (before V_b and after V_a) and the complex change definitions that will be evaluated for detecting respective instances (C). We focus on how nested complex change definitions are handled and how constraints are evaluated. In order to implement the language, we translate it into an already implemented language. As this approach concerns RDF data, we choose to rely on SPARQL, which provides similar capabilities to our language. The presented Algorithm involves two steps: the first step handles nested definitions, the second produces complex change instances.

As for the first step, suppose a complex change c whose definition is based on a set of complex changes ($D_C \neq \emptyset$). The detection of c instances depends on detecting the instances of each complex change in D_C and therefore follows their detection. Note that mutually dependent complex changes are not supported. In general, complex change

definitions constitute a directed acyclic graph, where nodes represent changes and edges dependencies between them. An edge departing from a complex change c arrives at changes in D_C according to its definition. Thus, detection follows a post-order depth-first scheme on the induced dependency graph by complex change definitions. This is stated in line 2 of proposed Algorithm. postOrderDfs function call runs over the set of complex changes C identifying the dependencies among changes, returning a queue Q of all changes in C, where the order of elements defines the order in which they have to be detected.

As for the second step, for each complex change c in Q, instances are computed (lines 3-10). The main idea is that our language is translated into SPARQL queries. Accordingly, simple and complex change instances and dataset versions are encoded as RDF data, so that constructed SPARQL queries are applied on them. Therefore, for each complex change an appropriate SPARQL query is created through createQuery function call (line 5). For this, changes in $D(c)$, constraints on their parameters and bindings are employed. Bindings indicate how to select complex change parameter values. Cardinality is taken into account to identify whether a change is optional. This query is executed on the detected change instances and dataset versions (line 6) in order to select change instances that verify the defined constraints. The query results are further elaborated, through createInstances function call (line 7), so that selected changes are grouped based on cardinality. Computed instances are added into the set of instances to be reported I (line 8, initialized in line 1), and are combined with simple change instances in order to be available for detecting depending complex change instances (line 9). Finally, the algorithm returns the set of detected complex change instances I (line 11).

Regarding query generation, testing value and relational constraints map to SPARQL filter expressions or nested queries with aggregation (in case they involve parameters evaluating into sets), while pre/post-conditions map to filter exists/not exists expressions over appropriate graphs holding the version before or after the change. Quantified expressions are also mapped to appropriate nested queries. Cardinality "?" and "*" map to optional declaration, indicating that respective changes may not be present. Bindings guide how query variables in select clause, representing complex change parameters, match query variables in where clause.

Algorithm: Complex Change Detection

Input: A set of complex changes C, a dataset version before V_b and after V_a, a set of simple change instances S_i

Output: A set of complex changes instances I of C

1 $I \leftarrow \{\}$;
2 queue $Q \leftarrow postOrderDfs(C)$; //complex changes are sorted following dependencies
3 **while** $!Q.isEmpty()$ **do**
4 $c \leftarrow Q.dequeue()$;
5 $query \leftarrow createQuery(D(c), F(c))$;
6 $resultSet \leftarrow exec(query, S_i, V_b, V_a)$;
7 $I_c \leftarrow createInstances(resultSet, F_C^{car}(c))$;
8 $I \leftarrow I \cup I_c$; //report instances
9 $S_i \leftarrow S_i \cup I_c$; // instances are available for detecting depending changes
10 **end while**
11 **return** ;

Regarding instance generation, the query results have to be iterated so that they are grouped appropriately given cardinality constraints for constructing complex change instances.

For example consider the following query, which corresponds to Add_Senior_Employee defined in Table 2. In the select clause we consider query variables corresponding to contained changes' identifiers (?c1, ?c2, ?c3, ?c4 and ?c5) and the values which will be assigned to the complex change instance parameters (?sx, ?m and ?x). In the where clause we consider the changes defined in change list and the constraints defined in filter list. For Delete_Reference and Move_Project _Assignment we use optional parts, due to "*" cardinality constraint. For post-conditions we use appropriate SPARQL filter expressions evaluating over the graph holding V_a. The first post-condition refers to Move_Project_Assignment and thus it is placed into the respective optional part. Also, it involves quantification, which is implemented through a nested query. The query results should be iterated for creating instances. Notice that Add_Employee and the first Add_Reference have cardinality equal to one. Thus, all rows having the same value in the respective query variables (?c1, ?c2) will form one complex change instance.

```
SELECT ?c1 ?sx ?c2 ?m ?c3 ?x ?c4 ?c5
WHERE { ?c1 rdf:type ch:Add_Employee;
ch:aep1 ?sx.
?c2 rdf:type ch:Add_Reference; ch:ar1
?sx; ch:ar2 ?m.
FILTER EXISTS {GRAPH <http://employeeVa>
{?m e:position e:Manager.}}
?c3 rdf:type ch:Add_Reference; ch:ar1 ?x;
ch:ar2 ?sx.
OPTIONAL {?c4 rdf:type ch:Delete_
Reference; ch:dr1 ?x; ch:dr2 ?psx.}
OPTIONAL {?c5 rdf:type ch:Move_Project_
Assignment; ch:mpap1 ?s; ch:mpap2 ?sx;
ch:mpap3 ?v.{SELECT ?c5 WHERE{?c5 rdf:type
```

```
ch:Move_Project_Assignment; ch:mpap1 ?s;
ch:mpap2 ?sx. FILTER NOT EXISTS {GRAPH
<http://employeeVa> {?s e:refersTo
?sx.}}}GROUP BY ?c5 HAVING(count(?s)=0)}}}
```

6 CONCLUSIONS

In this paper we argued that treating changes as first class citizens is a central issue in evolution management. This involves modelling, defining and detecting complex changes. Thus semantically rich changes and their interrelations are supported for interpreting evolution in multiple ways. We proposed our perception regarding complex changes, a declarative language for defining them on RDF(S) knowledge bases and a process for detecting complex change instances. Future work is directed in evaluating our approach in terms of language expressiveness and detection efficiency.

ACKNOWLEDGEMENTS

Supported by the EU-funded ICT project "DIACHRON" (agreement no 601043).

REFERENCES

Auer, S., H. Herre, 2007. A versioning and evolution framework for RDF knowledge bases. *In Perspectives of Systems Informatics.*

Berners-Lee, T., Connolly, D., 2004. Delta: An ontology for the distribution of differences between *RDF* graphs.http://www.w3.org/DesignIssues/Diff (version: 2006-05-12).

Franconi, E., Meyer, T., Varzinczak. I., 2010. *Semantic diff as the basis for knowledge base versioning.* In NMR.

Galani, T., Stavrakas, Y., Papastefanatos, G., Flouris, G., 2015. *Supporting Complex Changes in RDF(S) Knowledge Bases*. In MEPDaW-15.

Klein, M., 2004. Change management for distributed ontologies*. Ph.D. thesis*, Vrije University.

Noy, N.F., Musen, M., 2002. PromptDiff: *A fixed-point algorithm for comparing ontology versions*. In AAAI.

Papastefanatos, G., Stavrakas, Y., Galani, T., 2013. *Capturing the history and change structure of evolving data*. In DBKDA.

Papavasileiou, V., Flouris, G., Fundulaki, I., Kotzinos, D., Christophides, V., 2013. *High-level change detection in RDF(S) KBs*. In ACM Trans. Database Syst., 38(1).

Plessers, P., De Troyer, O., Casteleyn, S., 2007. Understanding ontology evolution*: A change detection approach. In J. Web Sem. 5(1)*: 39-49.

Roussakis, Y., Chrysakis, I., Stefanidis, K., Flouris, G., Stavrakas, Y., 2015. *A flexible framework for understanding the dynamics of evolving RDF datasets.* In ISWC.

Stojanovic, L., 2004. Methods and tools for ontology evolution. Ph.D. thesis, University of Karlsruhe.

Volkel, M., Winkler, W., Sure, Y., Kruk, S., Synak, M., 2005. SemVersion: *A versioning system for RDF* and ontologies. In ESWC.

Zeginis, D., Tzitzikas, Y., Christophides, V., 2011. *On computing deltas of RDF/S knowledge bases.* In ACM Transactions on the Web.

APPENDIX

Simple Changes on RDF(S) Knowledge Bases.
Add_Type_Class(a): Add object a of type rdfs:class. **Delete_Type_Class(a)**: Delete object a of type rdfs: class. **Rename_Class(a)**: Rename class a to b. **Merge_Classes(A, b)**: Merge classes contained in A into b. **Merge_Classes_ Into_Existing(A,b)**: Merge classes in A into b, b∈A. **Split_ Class(a,B)**: Split class a into classes contained in B. **Split_ Class_Into_Existing(a,B)**: Split class a into classes in B, a∈B. **Add_Type_Property(a)**: Add object a of type rdf:property. **Delete_Type_Property(a)**: Delete object a of type rdf:property. **Rename_Property(a,b)**: Rename property a to b. **Merge_Properties(A,b)**: Merge properties contained in A into b. **Merge_Properties_Into_Existing(A, b)**: Merge A into b, b∈A. **Split_Property(a,B)**: Split property a into properties contained in B. **Split_Property_ Into_Existing(a,B)**: Split a into properties in B, a∈B. **Add_ Type_Individual(a)**: Add object a of type rdfs:resource. **Delete_Type_Individual(a)**: Delete object a of type rdfs: resource. **Merge_Individuals(A,b)**: Merge individuals contained in A into b. **Merge_Individuals_Into_Existing (A,b)**: Merge A into b, b∈A. **Split_Individual(a,B)**: Split individual a into individuals in B. **Split_Individual_Into_ Existing(a,B)**: Split a into individuals in B, a∈B. **Add_ Superclass(a,b)**: Parent b of class a is added. **Delete_ Superclass(a,b)**: Parent b of class a is deleted. **Add_ Superproperty(a,b)**: Parent b of property a is added. **Delete_Superproperty(a,b)**: Parent b of property a is deleted. **Add_Type_To_Individual(a,b)**: Type b of individual a is added. **Delete_Type_From_Individual(a,b)**: Type b of individual a is deleted. **Add_Property_Instance (a_1,a_2,b)**: Add property instance of property b. **Delete_ Property_Instance(a_1,a_2,b)**: Delete instance of property b. **Add_Domain(a,b)**: Domain b of property a is added. **Delete_Domain(a,b)**: Domain b of property a is deleted. **Add_Range(a,b)**: Range b of property a is added. **Delete_ Range(a,b)**: Range b of property a is deleted. **Add_ Comment(a,b)**: Comment b of object a is added. **Delete_ Comment(a,b)**: Comment b of object a is deleted. **Change_ Comment(u,a,b)**: Change comment of resource u from a to b. **Add_Label(a,b)**: Label b of object a is added. **Delete_ Label(a,b)**: Label b of object a is deleted. **Change_ Label(u,a,b)**: Change label of resource u from a to b.

Software Crowdsourcing Challenges in the Brazilian IT Industry

Leticia Machado[1], Josiane Kroll[1], Rafael Prikladnicki[1], Cleidson R. B. de Souza[2] and Erran Carmel[3]
[1]*Pontifical Catholic University of Rio Grande do Sul (PUCRS), Porto Alegre, Brazil*
[2]*Department of Computing, (UFPA), Federal University of Pará, Belém, Brazil*
[3]*School of Business, American University, Washington DC, U.S.A.*
{leticia.machado.001, josiane.kroll}@acad.pucrs.br, rafael.prikladnicki@pucrs.br, cleidson.desouza@acm.org, carmel@american.edu

Keywords: Software Crowdsourcing, Software Development, Software Engineering, Brazil, Challenges, IT Industry.

Abstract: Software crowdsourcing has been regarded as a new paradigm for the provision of crowd-labor in software development tasks. Companies around the world adopt this paradigm to identify collective solutions to solve problems, ways to accelerate time-to-market, increase the quality and reduce the software cost. Although this paradigm is a trend in the software engineering area, several challenges are behind software crowdsourcing. In this study, we explore how the software crowdsourcing has been developed in the Brazilian IT industry. We have conducted 20 interviews with Brazilians practitioners in order to identify the main challenges for software crowdsourcing in Brazil. Additionally, we identified and discussed enablers and blockers' factors, practice implications and directions for future research in the area. Our paper aims to provide an overview of the software crowdsourcing in Brazil and motivation for researchers to better understand challenges faced by the Brazilian IT industry.

1 INTRODUCTION

Crowdsourcing (CS) is defined as the act of an organization to make its work available to an undefined, potentially large networked of people – a crowd - using an open call for participation (Howe, 2008). This concept has been adopted to disseminate corporate tasks that were traditionally performed by small groups of people.

CS has been adopted for several purposes such as innovative design (Howe, 2008), information peer production, knowledge and culture dissemination data analysis (Brabham, 2008), and software development (Lakhani et al., 2010; Wu et al., 2013).

CS in software development means to engage a global pool of online workers that can be tapped on-demand to provide software solutions or services (Lakhani et al., 2010; Stol and Fitzgerald, 2014).

Many computational platforms were created to handle the technical aspects of CS tasks, including broadcasting of tasks, task assignment, and submission and analysis of results (e.g. TopCoder, Crowdtest and WeDoLogos).

In Brazil, software CS is in the early stages. We have observed the lack of processes, models, and practices to support the Brazilian community. This is problematic because the adoption of CS in software development activities can help to increase Brazilian companies' competitiveness in the global software development market.

Our paper presents findings from a study conducted in the Brazilian IT industry. This study aimed to understand how software CS has been adopted in the Brazilian IT industry and the main challenges faced during its adoption. Additionally, we identified and discussed enablers and blocking factors for the software CS in Brazil, practice implications and directions for future research in the area.

Our study offers the following main contributions:

- **A Set of Challenges Concerning the Adoption of Software CS by the Brazilian IT Industry:** since Brazil is in the early stages of adoption of software CS, we explore the particular challenges associated with it practice in Brazil.
- **The Theoretical Foundation for Further Research in the Area and Developing of Solutions for the Brazilian Community:** to identify and understand which aspects are related to the adoption, or not, of software CS in the Brazilian IT industry is the first step towards integrating and facilitating the CS model in other organizations.

Machado, L., Kroll, J., Prikladnicki, R., Souza, C. and Carmel, E.
Software Crowdsourcing Challenges in the Brazilian IT Industry.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 482-489
ISBN: 978-989-758-187-8

2 SOFTWARE CROWDSOURCING

CS is a hybrid model regarding to intrinsic and extrinsic motivation (Mao et al., 2013). Motivation can be driven by financial rewards, which are extrinsic factors. On the other hand, crowd participants are also interested in reputation that can be earned through knowledge is shared – intrinsic intentions (Olson and Rosacker, 2013).

Despite this outward similarity, characteristics of service providers and suppliers are different in the two models. While with traditional outsourcing an entity subcontracts a handful of professional third-party companies, CS model turns to scale via an undefined, open, and heterogeneous online "crowd" to source in these needs (Saxton et al., 2013).

A significant distinction between software development strategies is the duplication of work. In CS, activities are performed in parallel, distributed in many chunks instead of single projects. The main differences among Innersourcing, Outsourcing, software CS, and Open Source Software (OSS) are presented in Table 1. The payment characteristics of software development strategies also are different. In software CS payments are based on reward per tasks (Âgerfalk et al., 2015). In OSS, knowledge is for sharing, with the focus on the development of better software and little if any attention given to profitability. Software CS is an application of the OSS principles to other industries. However, it receives an open and unidentified group that competes to solve a problem (Olson and Rosacker, 2013).

For each CS area three main elements are adopted as shown in Figure 1. The first component is the CS platform, which acts as the intermediates between the two other components and consolidates the tasks outcomes. The second component is the Crowd, which is globally dispersed. The third component is the Requesters. They are the companies or the individuals whom demands the work (tasks) (Prikladnicki et al., 2014).

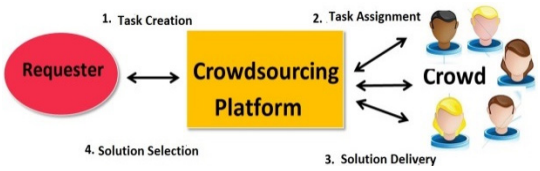


Figure 1: Basic crowdsourcing model.

Many authors argue that CS promotes creativity and problem solving (Kittur et al., 2013). However, software CS has many issues and unique features. Some of them still need for support (Wu et al, 2013): complex and heterogeneous tasks, interdependent tasks, several types of expertise, and activities of collective control.

These issues in software CS include quality, cost, diversity of solutions, delivery speed and, competitive scenarios. Furthermore, studies discuss many challenges and opportunities for better understand, evaluate and support the CS influence the software industry (Huhns et al., 2013).

2.1 The Brazilian IT Industry

Brazil is one of the largest economies in the world. Brazil has unique characteristics. Population of 200 million people; large and expanding domestic market; single language; sophisticated financial market; industry and business knowledge; qualified human resources; infrastructure; governmental support; favorable economic, political, and legal environment (Prikladnicki and Carmel, 2014). Thus, Brazil plays an important role in the global economy.

We decide to investigate software CS in the Brazilian IT industry because Brazil has the biggest and most diversified science, technology and innovation system of Latin America. According to Forbes magazine (2014), Brazil's economy outweighs that of all other South American countries, and Brazil is expanding its presence in world markets. We expect having emerging and large economies like Brazil taking place in the software CS market with more platforms, requesters

Table 1: Software Development Strategies.

	Innersourcing	**Outsourcing**	**CS**	**OSS**
Concept	Traditional business model	Traditional global business model	Web business model	Online Community
Nature of workforce	Specific group	Specific group	Open and undefined group	Open group
Incentives	Extrinsic	Extrinsic	Extrinsic/Intrinsic	Intrinsic
Intellectual Propriety	Enterprise	Enterprise	Winning solutions CS enterprise	Members' license
Payment	On payroll	Contract	Pay per tasks	Often unpaid

and large crowds.

The Brazilian IT industry does not take a risk before something (technology, model or process) is been widely known in other international markets. The Brazilian market has a more pre-cultural attitude in order to preserve conditions, principles, and existing processes. This market keeps a traditional way to carry out business. Few companies engage in innovative projects and pilot projects with a degree of uncertainty (Prikladnicki et al., 2014).

3 RELATED WORK

Whereas CS has been discussed in a wide variable domain, still have a small number of studies discussing software CS initiatives in different countries.

The adoption of CS platforms in South Africa is discussed by Chuene and Mtsweni (2015). This study was performed in order to understand how it is used and by whom. The authors report the embracing of CS initiatives has been slow, especially amongst public organizations, due to various reasons, such as lack of awareness. Some local CS platforms such as "Txteagle" launched in Kenya and Rwanda enables citizens to earn few dollars by completing micro-tasks on their mobile phones. A South African platform presented is focused on crowd-funding, social, and government crowdsourcing aspects. This study reveals a lack of information pertaining to the status and number of users benefiting from the adopted and/or deployed platforms.

In a recent study performed by To and Lai (2015), the authors discuss the latest developments in crowdsourcing in China. They describe CS scenarios in terms of concerns and opportunities. China offers CS advantages, it because includes the online population's size people's and their willingness to participate of the CS activities. Other opportunity was the Zhubajie's CS Chinese platform, where it services aren't closed off to English speakers. Chinese freelancers also participate in other markets. CS barriers reported in this study are the language between client and CS partners. Another concern was the censorship and government control, and intellectual property rights.

China and India are successful in attracting global outsourcing industry (Perera and Perera, 2014). China has a large educated workforce, high quality infrastructure, government keenness, etc. While India has lower labor cost, familiarity with western business practices, positive time zone difference, Indian owned global delivery centers and strong private-public partnership. In China, crowdsourcing platforms such as Zhubajie (zhubajie.com), TaskCN (taskcn.com) and K68 (k68.com) have attracted a lot of innovative talent and solution seekers, which greatly enhance business operations (Shao et al., 2012).

4 RESEARCH METHODOLOGY

An initial ad hoc literature review was carried out with the purpose of sharing the basic CS concepts with the research team and identifying the challenges to be addressed. Semi-structured interviews were conducted iteratively with Brazilians practitioners from different Brazilian companies. Our goal here is better understand how the software crowdsourcing has been adopted in the Brazilian IT industry. The interviews focused on both industry and academic perspective. Each interview lasted between 30 and 60 minutes.

We created an interview protocol with open-ended questions focusing on the discussions of both perspectives: *industry* - organizational motivations for leveraging the crowd, specific tasks to be completed and perceived impacts on the organization; and *academic* – sharing characteristics with related research areas like software testing, collaborative software engineering, distributed software development and distributed collaborative programming.

4.1 Participants

We conducted a total of 20 interviews, in which the majority of interviewees were males (18) and the others females (2). Thirteen participants were from the first group (industry) and 7 participants were from the second group (academic). These participants are mainly from the south of Brazil. Participants have 3 years of working experience in average.

The industry participants are classified under three different CS perspectives: buyer, platform, and crowd. We interviewed participants from two pioneer Brazilian CS platforms – Crowdtest and WeDoLogos. These companies are the two largest crowd testers and crowd designers in Latin America.

We interviewed academic participants during The Brazilian Conference on Software: Theory and Practice (CBSoft 2014). The CBSoft is one of the largest events held by the Brazilian Computer Society, with the goal of promoting and encouraging

Table 2: Participants' information.

Participant #	Job Title	Type of experience (Academic or Industry or Both)	Element
P1..P6	Tester	Industry and Academic	Crowd
P7, P8	Developer	Industry and Both	Crowd
P9..P11	Assistant Professor (System analyst)	Academic	Crowd
P12	Developer	Both	Requester
P13	Manager (IT Consultant)	Industry	Requester
P14	IT Manager (Host Service Company)	Industry	Requester
P15	IT Manager (E-Commerce Company)	Industry	Requester
P16	Manager (IT Company)	Industry	Requester
P17	Manager (Media Company)	Industry	Requester
P18	CEO (Innovation Consultant)	Industry	Requester
P19	CEO (WeDoLogos)	Industry	Platform
P20	CEO (Crowdtest.me)	Industry	Platform

the exchange of experiences between the scientific, academic and professional communities in Software Engineering (SE). The participants' details are presented in Table 2. The last column called Element presents the three basic elements of the CS model, in which each participant was classified (see Figure 1).

4.2 Procedure

We conducted the interviews face-to-face, by voice or video conference call, and by email. Some conversations were not audio recorded because of companies' confidentiality issues. The interviewees were asked to report their experiences in software CS under six aspects: (1) CS initiative, (2) CS platforms, (3) CS tasks and projects, (4) CS payment, (5) business impact, and (6) the future. We present the interview questionnaire in Table 3.

Table 3: Interview Questionnaire.

Aspects	Questions
CS Initiatives	Do you know CS? Tell us about your CS experience?
	Are you doing micro-tasking specifically? Or are you doing macro task projects?
CS Platforms	Which Platforms (middlemen) have been used?
	What is the number of workers in the crowd?
CS Tasks and Projects	What have you done to achieve and inspect for quality? What has worked best?
	How do you manage day-to-day tasks?
CS Payment	Is the enterprise encouraging / discouraging the use of paid CS?
Business impact	By what measure was it successful? What has made this challenge a success?
	How did you measure success?
Future	What are your plans for CS?
	What is the Brazilian CS scenario for the next three years?

4.3 Data Analysis

Our data analysis was guided by techniques associated with less procedural versions of the grounded theory (GT). Specifically, we applied the techniques of coding and constant comparison as recommended by Corbin and Strauss (2008). These techniques helped us to elicit emergent themes in the Brazilian IT industry, to identify concepts in the collected data and to link these concepts to higher-level categories.

5 FINDINGS

Our findings show that both academic and industry participants have different experiences using CS for micro and complex tasks. They have adopted CS for software and other domains. They have performed tasks such as, testing service and image recognition.

We also identified collaborative tools adopted by Brazilians to improve software development. These tools are based on crowd knowledge such as GitHub and Stackoverflow.

Table 4: CS research in Brazil.

Research Area	Crowdsourcing Topics
Experimental SE Software Engineering	Barriers to contribute to the open sourcing process
Software testing	Crowd testing model of the enterprise
Collaborative Software Engineering	Collaborative tools – CS Platform Crowd knowledge
Software Ecosystems Platform	Distributed software development and open participation (crowd)
Distributed collaborative programming	Motivation, coordination and sharing knowledge

Researchers are investigating Crowd testing in two contexts – distributed and traditional testing software. We observed from these findings research opportunities in other software ecosystems that share the same characteristics with other research areas. Table 4 presents the research areas and topics in software CS.

5.1 CS Elements Perspective

Our findings show that Brazil has a few CS buyer's initiatives and platforms. On the other hand, Brazilian active members' are emerging both on national and international CS platforms.

We describe our findings based on three perspectives: Crowd, Requesters, and Platform. These results can be categorized in enablers and blockers' factors. Table 5 shows these factors.

Table 5: Enablers and Blockers' factors in Software CS.

Elements	Enablers	Blockers
Crowd	Extra money	Poor feedback
	Shared knowledge	Few collaboration
	Curiosity	Scarce context project information
	Free time	Unavailability of documentation
Requesters	Scalability	Specific business knowledge
	Save money and time	Low quality of services
	Creativity	Maturity of suppliers (crowd)
	New ways to do the same	Identifying a specific process
Platform	Fast delivery	Data confidentiality
	Reduced cost	Very specific business rules
	Diverse types of testing	Laws and taxes involved

An enabler factor means a characteristic that promotes or motivates the CS practice. On the other hand, a blocker factor means a characteristic that inhibits or limits the CS practice.

As enablers, there is the collective intelligence of the software engineering industry with more diversity, creativity and knowledge sharing. Scalability, cost reduction and time-to-market are also important enablers for platforms and requesters. As blockers, the requesters pointed out the low quality of services, the difficulty in identifying a specific process to distribute tasks to the crowd and the maturity and adoption of CS in Brazil.

5.2 Challenges

We identified challenges related to three areas: Tasks, Processes and People. Each area can describe one or more challenges.

- **Tasks – Lack of Quality**

Challenges related to tasks area includes the lack the quality of platforms and micro-tasks. The platform should provide clear information and support documentation to the crowd, appropriate structure for the submission of solutions, and feedback to submitted tasks.

Micro-tasks refer to the personal demand created by requesters. The configuration of the micro-task request is performed through CS parameters such as specific subject, constraints, quality issues, monetary reward, and target worker. Participants reported the unavailability of documentation on tasks requirements, specific business rules and scarce context about the tasks. In some projects, participants report the need for more information to complete and deliver distributed tasks. They also need to achieve the requesters' expectations. Stol and Fitzgerald (2014) also describe the task quality assurance challenge under the requester perspective.

Testers describe the lack of information on reported bugs. A tester participant, who has worked on the Crowdtest Brazilian platform, describes how errors are reported.

"Everyone reports errors in the same place and the tool does not return with the reason why the reported error was not considered as an error to the platform (client), and we were no guarantee" (Participant 5).

On the other hand, the requester describes the quality of crowd deliveries. In his opinion, the lack of qualified works is the main problem.

"Sometimes there is a lack of professionalism in this environment. Most of the time filters are not performed by the platforms to allocate the profile of qualified members" (Participant 14).

A buyer, who has bought services on WeDoLogos platform since 2010, says that in this field there are few professionals with specific skills. For him, the majority of workers are not skilled. Although the quality of the service is considered good for him, he receives few proposals to perform the work (approximately 10 proposals per project). On the other hand, another participant describe a project in which he received more than 200 proposals. However, the quality of the delivery by the crowd was very low. Only 10% of solutions received from the crowd could be take. For some projects, he has considered to return to the

traditional model of hiring service to meet his needs.

"I believe that for small businesses the crowdsourcing model can work. The quality of service is low and any delivery is part of the crowd" (Participant 12).

- **Processes – Lack of CS Processes**

Since the crowd and requesters from Brazil have little experience in software CS, we observed challenges related to the immature adoption of CS processes. Users are not familiar themselves with the CS processes.

Participants from the buyer's group report the importance of having a process to support the adoption of CS initiatives. Participants emphasize limitations to adopt CS models in business. These limitations are related to CS management processes in terms of collectively coordination, communication and collaboration.

"Brazil is a very conservative country and it needs to prepare people to work with crowdsourcing. It is necessary to have a strong process behind the platform and people to support business" (Participant 13).

"Crowdsourcing is difficult in our company because it is necessary to have a visibility of tasks (progress activities, for instance "to do", "doing" and "done"). We have a strong work process orientated on quality and productivity. Crowdsourcing could be a new direction to the future but it requires a maturity level and another mindset" (Participant 16).

A participant reports the use of CS model for open innovation domain in Brazil. This project started in 2006. Recently, this project has a partnership with a CS innovation platform called *Innocentive*. The requester describes how the CS processes work in this project.

"I believe in the crowdsourcing model for my business and my clients. The process created by them has been used with good results for the clients. The success in crowdsourcing projects depends on some factors, such as number of projects, participant engagement, ideas proposed by participants, deployment of solutions, number of participants, and concept of phases to crowdsourcing projects (dependent on the complexity and investment by client)" (Participant 18).

The participant also describes tools to support the process. These tools are only adopted as an interface between customers and network (crowd). For him. It is more important to focus on how the project design is planned and executed by the workers.

- **People – Cultural Barriers**

Brazil has particular laws and legislation. The country is very concern on labor law and bureaucratic issues.

Brazil is a conservative country in terms of distributed software development. Usually, Brazil receives a lot of outsourcing demands but it is not used to outsource. The most of the time, Brazil is much more a supplier than a consumer.

The country is a special case in terms of software development. Brazilians companies and labor participate in global CS marketplaces, but it also "plays in their own sandbox". It may happen because of language issues. Portuguese is the official language in Brazil and the majority of the population do not speaks another language.

Another fact is that Brazilians prefer to have a permanent job besides to be to have freelancer job. In addition, participants report do not trust in having a virtual contract of work. They are very conservative people.

One of our findings shows a contradiction between the collaborative culture of Brazil and the competitive environment in the economy. Brazil still trying to introduce a new model of work.

"Companies control logic and create an antagonism in relation to digital networks. Companies have difficulty in changing their business to work in a complex world. Companies want the control of their idea" (Participant 8).

Brazilians companies do not understand the distributed workforce within a collaborative system. They have concerns about intellectual property and business rules. According to the participant 18, CS is much more talked about than consumed in Brazil.

6 DISCUSSION

CS is an emerging topic in software industry. It provides a new approach for companies involve their workers with innovation activities. However, despite the positive effects, many challenges are identified for CS practice. The Brazilian IT industry has specific challenges that make this country different from others.

We found that the main challenges faced by Brazilians practitioners are concentrate in three areas Tasks, Processes, and People. We also found eleven enablers and blockers factors for the CS practice in Brazil.

Tasks are difficult to manage in CS environments. Requesters expect to receive a task with certain level of quality. However, in some cases

the delivery do not attend the expectations of the requester. According to (Li et al., 2013), one of the most problems in CS is quality control to ensure the quality of results. The factors of quality for CS tasks are the number of participants, tasks assignment to workers according to their individual expertise, and the reward amount. The inappropriate worker-task matching may harm the quality of the software deliverables (Mao et al., 2015).

On the other hand, workers report the lack of information that can result in a task delivered with low quality. When workers understand what information is need for the task specification, it will be possible to provide solutions to problems that meet customers' needs. However, to Wu et al. (2013) the vendor selection has a direct correlation with the quality of an outcome. Workers are attracted by an open call format rather than being selected. It encourage the non-skilled workers to participate. The list of countries with higher level of active members shows Brazil on 14º ranking position between 50 countries. Country rankings are based on an aggregation of the TopCoder members within a particular country that have competed within the last 180 days (https://community.topcoder.com/).

The lack of processes definition is another challenge faced by the Brazilian IT industry. To take advantage the power of software CS, it is important to define the proprieties, elements, responsibilities and interaction flows of software CS as a new software development process (Kittur et al., 2013). While other countries like United States adopt and invest in crowdsourced development processes, Brazil adopts a timid posture regards to it. Brazil has only two CS platforms to support software activities. We believe that online markets for software CS tasks such as software project development activities, still have not received attention from companies and workers. Currently, Brazilian platforms do not meet the requesters' expectations. According to our findings, Brazilian platforms support only few types of activities.

Portuguese is the official language in Brazil. English is the global language of business. The majority of Brazilians speak only Portuguese. Thus, the Brazilian community face difficulties to use international CS platforms due to the language.

The intellectual propriety in software CS is a world polemic question. In Brazil, this question is amplified because the Brazilian laws and legislation have characteristics of trade protectionism.

In literature, few authors explore region-specific practices in CS software project. Europe and United States are well populated with CS participants, but that still does not say much about potential differences in acceptance of CS across the globe.

In our study, some cultural aspects in CS are presented. Brazilians are highly creative in their own way, but the country is still underdeveloped in terms of software CS. Cross-cultural differences in the adoption of CS and open approaches to business are still under-explored.

CS is a business concept that focus on the use of intelligence, collectivity and volunteer knowledge to solve problems, improve or develop new products, technologies and services (Brabham, 2008). Nevertheless, CS still not clearly understood by many companies that can take advantage of this concept. Brazilians are highly creative in their own way, but the country is still underdeveloped in terms of software CS. Cross-cultural differences in the adoption of CS and open approaches to business are still under-explored.

Under the Brazilian perspective, there are many issues regards to CS elements' (requester, platform and crowd). To Carmel and Eisenberg (2006), Brazilian national software builds pride inside the Brazilian software community to develop software under conditions of hardship. For these authors, Brazilian software companies do not believe in its capacity to create and offer job for other workers and other markets.

Every country is unique and has its own specific challenges when it comes to change the way of work, like implementing software CS. This study gives a starting point on region-specific practices in crowdsourced software development.

6.1 Limitations of this Study

We are aware of the limitations of this study, since our study does not seek to establish any causal relationships, we do not discuss threats to internal validity.

The qualitative analysis of the interviews was performed by the authors together, which limited the effects of possible researcher bias in the analysis. We also adopted grounded theory to analyze collected data using descriptive statistics and techniques (Corbin and Strauss, 2008).

We interviewed Brazilians practitioners with different experience levels. The imbalance experience could have influenced positively or negatively in our findings. Unfortunately, the identified findings are not exhaustive. They only represent those that have been experienced and observed by our participants. We have carefully

selected the participants in this study.

7 CONCLUSIONS AND FUTURE WORK

In this study, we investigate how the software CS has been developed in the Brazilian IT industry. We found that Brazil is very conservative and moderate in terms of adoption of software CS in IT scenario. Brazil has a few requesters' demand, both in national and international CS platforms. Also, we found few CS platforms to support the Brazilian market.

Although Brazil is an important software market and one of the most important emerging economies, we are not surprised with our findings. Brazil has a weak participation in software CS initiatives.

Our results show that the main challenges in software CS in Brazil are related to Tasks, Processes, and People. We believe that for the software CS to work effectively, it is important to better understand the issues related to the three CS elements (Crowd, Requesters, Platform). Given this perspective, the research we present here is of value to both to academic and industrial communities. We also believe that these findings are particularly important from the Brazilian perspective; however, they also help add to the body of evidence in the field of software engineering.

In spite of the challenges, we believe that CS will get new labor markets in future. Markets that are disrupted, like the Brazilian software market, shall see changes in the types of tasks that are currently being performed. Also, software development though CS may it will help to alleviate the Brazilian cultural limitations mentioned in this study.

More empirical research is needed on how to develop software CS in Brazil. We plan to follow our case organizations to see how they minimize the identified challenges, collecting more detailed data about their software practices, as well as by additional interviews.

ACKNOWLEDGMENTS

This work is partially funded by CNPq (312127/2015-4).

REFERENCES

Âgerfalk, P. J., Fitzgerald, B., and Stol, K. J., 2015. Software Sourcing in the Age of Open: Leveraging the Unknown Workfoce. *Springer*.

Brabham, D. C., 2008. Crowdsourcing as a Model for Problem Solving: An Introduction and Cases. In *Convergence*, 14, 1.

Carmel, E., and Eisenberg, J., 2006. Narratives that software nations tell themselves: An exploration and taxonomy. *Communications of the Association for Information Systems, 17(1), 39*.

Chuene, D.; Mtsweni, J., 2015. "The adoption of crowdsourcing platforms in South Africa," in *IST-Africa Conference, 2015*, pp.1-9.

Corbin, J. M. and Strauss, A. L., 2008. Basics of qualitative research: Techniques and procedures for developing grounded theory, *3rd edition. Sage*.

Forbes Magazine, Brazil: Profile. Available at http://www.forbes.com/places/brazil/. Access on November 2015.

Howe, J., 2008. Crowdsourcing: Why the Power of the Crowd Is Driving the Future of Business. In *Crown Business*.

Kittur, A., Nickerson, J. V., Bernstein, M. S., Gerber, E. M., Shaw, A., Zimmerman, J., Lease, M. and Horton, J. J., 2013. The Future of Crowd Work. In *Proc. CSCW. ACM.*

Lakhani, Karim, David A. Garvin, and Eric Lonstein, 2010. TopCoder (A): Developing software through crowdsourcing. *Harvard Business School General Management Unit Case* 610-032.

LaToza, T.D., Towne, W.B., van der Hoek, A. and Herbsleb, J.D., 2013. Crowd Development. Proc. 6th CHASE Workshop. San Francisco, CA.

Li, K., Xiao, J., Wang, Y., and Wang, Q., 2013. Analysis of the key factors for software quality in crowdsourcing development: an empirical study on TopCoder.com. In *37th Annual Computer Software and Applications Conference, pp. 812-817.*

Mao, Ke, et al., 2015. A Survey of the Use of Crowdsourcing in Software Engineering. RN 15:01.

Olson, D. L., and Rosacker, K., 2013. Crowdsourcing and open source software participation. *Service Business, 7(4), 499-511*.

Prikladnicki, R., Machado, L., Carmel, E., & de Souza, C. R., 2014. Brazil software crowdsourcing: a first step in a multi-year study. In *1st International Workshop on Crowdsourcing in Software Engineering(pp. 1-4).*

Saxton, G., Oh, O., Kishore, R., 2013. Rules of Crowdsourcing: Models, Issues, and Systems of Control. *Information Systems Management* 30, 1–13.

Stol, K. J., and Fitzgerald, B., 2014. Two's company, three's a crowd: a case study of crowdsourcing software development. In *Proceedings of the 36th International Conference on Software Engineering (pp. 187-198). ACM.*

To, Wai-Ming, and Linda SL Lai., 2015. Crowdsourcing in China: Opportunities and Concerns." *IT Professional* 17.3: 53-59.

Wu, Wenjun, Wei-Tek Tsai, and Wei Li., 2013. An evaluation framework for software crowdsourcing. *Frontiers of Computer Science Journal* 7, no. 5: 694-709. Publisher Springer Berlin Heidelberg.

Investigating the Adoption of Agile Practices in Mobile Application Development

Alan Santos[1], Josiane Kroll[1], Afonso Sales[1], Paulo Fernandes[1] and Daniel Wildt[2]
[1]*Computer Science Department, Pontifical University Catholic of Rio Grande do Sul (PUCRS), Porto Alegre, RS, Brazil*
[2]*WildTech, Porto Alegre, RS, Brazil*
{*alan.ricardo, josiane.kroll*}*@acad.pucrs.br,* {*afonso.sales, paulo.fernandes*}*@pucrs.br, dwildt@wildtech.com.br*

Keywords: Software Engineering, Mobile Application, Software Development, Agile Practices, Challenges, Benefits.

Abstract: The mobile application development market has been dramatically growing in the last few years as the complexity of its applications and speed of software development process. These changes in the mobile development market require a rethinking on the way the software development should be performed by teams. In order to better understand how agile practices support mobile application development, we applied a questionnaire to 20 undergraduate students. These students have been training in an iOS development course combined with agile practices. Our study aims to identify challenges and to report the students experience on the adoption of agile practices to develop mobile applications. Our findings reveal that agile practices help mobile software development mainly in terms of project management and control and development speed. However, aspects of user interface and user experience, different development platforms, and users expectations still point challenges in developing mobile applications.

1 INTRODUCTION

Mobile application development is a new trend in the software industry. It also plays an important role in the economic development of a country as well as in teaching and learning (Zhang, 2015). The combination of devices such as cameras, sensors, touch and GPS with mobile platforms increase the possibilities for developing new mobile applications (apps). Additionally, devices have become more complex and mission critical (Lewis et al., 2013) due to the sudden wave of mobile device use.

According to Wasserman (Wasserman, 2010), mobile devices have been adopted in different ways for desktop or laptop computers. Mobile applications development can be similar to software engineering for other embedded applications. However, mobile applications development present some additional requirements that are less commonly found if compared to traditional software applications. The relevance of mobile software products has reached a point in which its devices have become one of the most popular platforms for the distribution and use of user-oriented software (Corral, 2012). The development speed in mobile software development has become a key factor due to developers' possibility of submitting applications (apps) directly to the market. Thus, it is necessary to identify agile practices to implement mobile applications as well as to provide a good learning experience.

In this paper, we investigate challenges in mobile application development and the students' experience on the adoption of agile practices for developing mobile applications. In order to achieve this goal, we applied a questionnaire to 20 undergraduate students who have been attending an iOS development course. This course adopted agile practices to develop different types of mobile applications. Our results describe the participants' perception on the use of agile practices, challenges, and perceived benefits. The main contribution of this paper is to provide a further discussion about the adoption of agile practices for mobile application development.

The remainder of this paper is organized as follows: Section 2 introduces a brief background about mobile application development while Section 3 presents a background on agile software development. In Section 4, we describe the research methodology adopted in this study and, in Section 5, we present the results. Section 6 discusses our results. Finally, we draw our conclusion an future work in Section 7.

Santos, A., Kroll, J., Sales, A., Fernandes, P. and Wildt, D.
Investigating the Adoption of Agile Practices in Mobile Application Development.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 490-497
ISBN: 978-989-758-187-8

2 MOBILE APPLICATION DEVELOPMENT

Since 2008, Apple and Google have opened their application store for iOS and Android platforms, a point where mobile apps started quickly evolve. Mobile application development is a process in which applications are developed for small handheld devices, being either pre-installed on devices during manufacture or downloaded from application stores or other software distribution platforms (Flora and Chande, 2013). Following the evolution of mobile application development, the traditional software development life cycle is no longer the only approach because long project planning phases and long development cycles can result on outdated mobile applications.

There are different programming environments available for the major mobile platforms (Wasserman, 2010), for Windows Phone there is Microsoft's Visual Studio environment, for Android platform there are Android development tools plug-in for Eclipse, and Apple iOS Dev Center has the Xcode package. According to Xanthopoulos and Xinogalos (Xanthopoulos and Xinogalos, 2013) with the currently increasing number of mobile platforms, developing mobile applications has become difficult for companies, as they need to develop the same applications for each target platform. The typical process for developing native applications is the most appropriate way of deploying mobile apps but it has one major disadvantage: it is not possible to reuse the source code for another platform; the same app must be redeveloped from the beginning.

Mobile web applications are mainly based on technologies such as HTML and JavaScript and do not require installation or device upgrades, enabling information processing functions to be initiated remotely on Web server (Huy and vanThanh, 2012). Some of the web applications drawbacks are: limited access to the underlying device as hardware and data and the extra time needed to render the web content (Xanthopoulos and Xinogalos, 2013). Hybrid development is another approach to develop a mobile application which tries to combine the advantages of web and native apps where applications are primarily built using HTML5 and JavaScript, and a deep knowledge of the target platform is not required (Xanthopoulos and Xinogalos, 2013). According to Alston (Alston, 2012), many mobile applications that are developed are considered to be alternative applications. These applications are developed for a specific platform and it have access to the hardware of a device through the use of Application Programming Interfaces (APIs).

The adoption of a suitable software development methodology is very important in mobile software engineering, since software applications are changing and evolving all the time based on immediate user requirements (Kaleel and Harishankar, 2013). Authors describe Scrum practices as the best suit requirements of android software development and applied them in designing a mobile software development methodology where they were able to successfully develop a secure backup application using important features from Scrum methodology such as adaptability to evolving requirements, technically strong development teams and effective communication through daily meetings (Kaleel and Harishankar, 2013).

3 AGILE DEVELOPMENT

Agile development or adaptive development are aimed to rapidly adapt to the changing reality. An agile method emphasizes communication and collaboration in an iterative process (Smite et al., 2010).

The adoption of agile development makes software processes more flexible, helps in continue learning and incremental delivery, quickly and easily adapting to requirements and technologies changes. Moreover, agile development focuses more on the human aspects of software engineering than the processes, human interaction over tools and processes (Flora and Chande, 2013). Authors also performed a review and analysis on mobile application development process using agile methodologies. According to authors agile development has fit for mobile application development. In this context, there are studies which recommended that agile practices are a good choice and assures different phases of software development life cycle to solve the mobile application development issues (Flora and Chande, 2013), they evaluated the following mobile development process: Mobile D, RaPiD 7, Hybrid Methodology Design, MASAM and SLeSS where they found that work related to mobile software confirms agile practices to be a natural fit for the development of mobile applications and an appropriate agile method could be selected for a given project and can be tailored to a specific requirement based upon project's complexity and team size.

Agile development is recommended to small-to-medium-sized projects, software development organizations are increasingly recognizing the need for agility.

In literature, Extreme Programming (XP) and Scrum are the most common agile methods for mobile application development. According to Paasivaara et

al. (Paasivaara et al., 2008) these methods can be easily customized by software companies. We describe these agile methods and others in the following subsections.

3.1 Extreme Programming

Extreme Programming (XP) is a discipline of software development which emphasizes productivity, flexibility, informality, teamwork, and the limited use of technology outside of programming, working in short cycles and every cycle starts by choosing a subset of requirements from a larger set (Macias et al., 2003).

According to Moore and Flannery (Moore and Flannery, 2007), XP implements a groupware style development where feedback is obtained by daily testing the software where developers deliver the system to the customers as early as possible, allowing a rapid response for requirements and technologies changes. Beck (Beck, 2000) present XP as a light-weight methodology for small-to-medium-sized teams developing software in the face of vague or rapidly-changing requirements.

3.2 Scrum

Scrum is an iterative and incremental agile software development approach. It offers a framework and set of practices that keep everything visible, allowing practitioners to know exactly what is going on and to make adjustments in order to have the project moving towards desired goals. The adoption of Scrum practices is the main factor to successfully develop software projects (Scharff and Verma, 2010).

The scrum workflow is a sequence of iterations called *sprints* which have a duration between one and four weeks each. The team has the work foundation as part of a product backlog which is a list of requirements and priorities.

Each sprint has daily meetings where each team member answers what he/she has been done on the previous day, what is going to be done in the current day and if there is any roadblock to move forward on development activities. At the end of each sprint there is a product demo called *Sprint Review* and after that it is handled a lessons learned session called *Sprint Retrospective* (Reichlmayr, 2011).

4 RESEARCH METHODOLOGY

We applied a questionnaire to a group of 20 students from the iOS development training course. This course is provide for a large software company in order to train undergraduate students on mobile application development for iOS. The course takes 4 months duration.

In this study, we selected 20 from 87 students, who were attending the course. We adopted a random selection to obtain a pool of participants.

During the course, each student has his/her own equipment to use as part of the class meetings and projects and worked in teams from two to five individuals. The course curriculum includes the following subjects: Object-Oriented Programming, User Interface (UI) components, Model View Controller, Data sources, Navigation, Animations and Frameworks. The course also covered an introduction to Scrum framework. After taking theoretical lessons, all students work for four months to develop real mobile applications using agile practices to support it.

The participants are on average at the 5th semester and majority of participants who answered the questionnaire are from an IT related field: 30% from Computer Science, 35% from Information Systems, 10% from Computer Engineering, 10% from Systems Analysis and 15% from Other courses.

Another profile information from the overall 87 students attending the training is that 35% of the students already had previous software development courses using Java and C#. In this context, 68% had up to 3 years of experience in development, 18% had between 3 and 5 years of experience, and 14% had more than 5 years software development experience. Only 10% of the students had a previous contact with mobile application development. Most of previous students experience were from other courses, as well as from the industry. In the software development methodology analysis, 65% did not have any previous contact with software development methodologies, 20% had previous contact with some practices of agile development, and 15% had contact with traditional software development approach. Table 1 present the participants information.

The course is facilitated by 6 instructors with experience in iOS development, academic and project management background. Four of them, have more than five years of experience as software developers. The course combines elements of Challenge-Based Learning (CBL) and Scrum in order to help the students to develop their apps (Santos et al., 2015).

At the end of the training course, we applied a questionnaire with eight research questions. Six questions to collect the background information of the participants (Name, Age, Undergraduate course, Semester, Previous working/study experience in agile practices, Previous working/study experience in mo-

Table 1: Participants information.

Participant	Age	Course	Semester
A	24	Computer Science	8
B	23	Information Systems	6
C	21	Computer Engineering	5
D	22	Information Systems	7
E	27	Information Systems	5
F	21	Information Systems	4
G	24	Computer Science	3
H	22	Information Systems	5
I	19	Information Systems	4
J	21	Computer Engineering	5
L	34	Systems Analysis	3
M	19	Information Systems	5
N	20	Computer Science	4
O	20	Computer Science	3
P	26	Computer Science	4
Q	20	Business	5
R	24	Systems Analysis	3
S	21	Engineering	9
T	22	Systems Analysis	7
U	24	Computer Science	5

bile development). The other two questions related to the adoption of agile practices to develop mobile applications. The following questions are presented:

- Q1: *What are the challenges in mobile application development?*
- Q2: *What is your opinion about the adoption of agile practices for mobile application development?*

5 RESULTS

The following subsections outlines the results related to the research questions related to the adoption of agile practices to develop mobile applications. We adopted the content analysis as a qualitative research technique to identify the challenges and perceived benefits on the adoption of agile methods for developing mobile applications.

5.1 Challenges in Mobile Application Development

In mobile application development, apart from adopt agile or a traditional approach, developers face many challenges. Based on our data collection, we identified five main challenges related to the adoption of agile practices in mobile application development. Table 2 shows these challenges.

- **Define UI/UX (User Interface/User Experience Design):** UX was cited as one of the factors that differ developing mobile applications for traditional applications. This is point as a challenge because of the diversity of devices, sensors and features that may be are utilized using a mobile device. UI has also been cited as one of the factors that differ developing mobile applications for traditional applications. It due to the diversity of devices and different sizes and development platforms that can be used to develop applications. Participants explain this challenge.

  > *I think the main difference is about UI, not for the huge amount of different screen sizes, but the way applications are used on a desktop computer was always using a keyboard and mouse as input. We have a keyboard when using mobile devices, but instead of the mouse we have touch screen, that has numerous other representations to click, not to mention the use of all other sensors available, which makes creating the interface to integrate harmoniously challenging.* (Participant B)

  > *In my opinion, it's different because you need to think much more in the user experience. Usually the applications are for a general audience, then you should pay attention to all aspects (accessibility, design).* (Participant E)

Table 2: Challenges for mobile application development.

Challenges	Frequency
Define UI/UX	50%
Different users' expectations	30%
Different development platforms	20%
Continuous update	10%
Devices and applications performance	10%

- **Different Users' Expectations:** the diversity of users and their expectations is identified as a challenge in mobile application development. First, a single application may have millions of users, according to the sense that a lot of users also corresponds to a large diversity of users, with different expectations, demands and devices. Another point raised it is also the question of the speed in which mobile solutions need to be released.

  > *I think the main difference to develop mobile (applications) over other platforms is the proximity to the user. It is common for a mobile application to be used by millions of people, while a desktop system is different. In my point of view, mobile applications can help change the lives of people in a more direct and fast way, compared to some other systems. The biggest challenge is to promote solutions that really make a difference in people's lives.* (Participant J)

- **Different Development Platforms:** differences between hardware and software platforms have also been identified as one of the differences and challenges in developing applications for mobile devices. It due to the fact that the amount of application program interfaces (APIs) in each of the development platforms, as well as different features and differences in hardware.

  The great diversity of types and capabilities of these devices also creates a challenge for developers, because they need to develop the system in such a way that it is able to run satisfactorily in a wide range of devices. (Participant K)

- **Continuous Update:** the constant updating of technologies is also cited as one of the main challenges due to frequent updating of development platforms, as well as the frequent launch of devices with different sizes and features. A participant describe it.

  As challenges, I believe the fact that you have to keep up to date because the mobile development is always emerging innovations, new frameworks, new languages. (Participant C)

- **Devices and Application Performance:** another issue reported by the participants is performance on data access. It happens because of hardware limitations. We can also observe this aspect as an important aspect on the mobile development based on the following answers from the participants.

  It's different, because we have to think of something practical that fits in a relatively small screen and that is attractive. I think the biggest challenge is to be always updated and seek the best performance for the application, or it will become obsolete very quickly. (Participant B)

5.2 Perceived Benefits of Agile Practices for Developing Mobile Applications

Agile development as well mobile application development are research areas with many important aspects to be investigated. Despite of its challenges, we also identified a set of eight benefits of the adoption agile practices for developing mobile applications. Table 3 list the benefits.

- **Improves the Management and Control:** agile process address the inherent problems of traditional development using product demand and delivery, and also control of ongoing projects. Thus, agile processes implement control through frequent inspection and adaptation and support the project management.

  I believe it is extremely important, it enables better organization and control of tasks as better ways to follow the team. (Participant H)

Table 3: Perceived benefits of agile development for developing mobile applications.

Benefits	Frequency
Improves the management and control	45%
Improves development speed	25%
Continuous improvement	15%
Promotes a life-cycle delivery	15%
Support multiple interactions	10%
Improves communication	10%
Improves performance	5%
Allows transparency	5%

- **Improves Development Speed:** agile practices helps to attain development velocity. It specially because agile practices focus on short development cycles. Agile development teams tasked to deliver high-value features quickly.

  Agile practices positively influence the mobile development, because they are usually solutions that require immediate and rapid development. With many interactions agile is fundamental because with this the team is able to design and prototype a product with more speed, unlike other methodologies. For example, Waterfall approach validates the implementation only at the end of the cycle. (Participant C)

- **Continuous Improvement:** agile principles, practices, and methods support continuous improvement. Through constant iterations, iterative planning and review, agile development brings the expected results.

  The use of agile practices helps to make application development safety because it is possible to identify and eliminate failures or unwanted behaviors quickly and accurately. (Participant M)

- **Promotes a Life-cycle Delivery:** one of the great advantages of agile software development is the wealth of practices, techniques, and strategies that promote a delivery life-cycle. Agile teams will adopt a life-cycle that is the most appropriate for their situation. The delivery life-cycle is goal-driven.

  I believe that the use of agile methods help one mobile team to organize and deliver. Es-

pecially if the project is very long. This requires collecting metrics during the iterations. (Participant C)

- **Support Multiple Interactions:** the product lifecycle goes from the initial idea for the product, through delivery, to operations and support and often has many iterations of the delivery lifecycle. Multiple iterations promote fast development cycles and incremental improvement of applications. Furthermore, multiple iterations allow teams not only to plan at the iteration level but also to conduct long term release planning (Smite et al., 2010).

  I believe it is fundamental for development, mainly by constant reviews that facilitate troubleshooting and redefinition of the scope of the project (if needed). The various existing iterations on agile methods are extremely important for the application's success. Agile methods facilitate and assist mobile development. (Participant E)

- **Improves Communication:** agile development in general use a set of values, principles and practices to guide teams in being as agile as possible. It includes the adoption of models to support communication and understanding. Their adoption facilitates communication between the group and make team members more critical.

  It improves the communication and teamwork among team members providing a realistic view about project progress. (Participant R)

- **Improves Performance:** agile practices can help to improve the project performance in mobile development environments. It because agile practices provide a major performance of developers. Agile teams provide an agile plan with progress updated every day.

  I think that helps a lot in performance improvement. Perhaps even more than other areas of development. It fits very well with mobile development. (Participant A)

- **Allows Transparency:** it was also raised as a point of clarity and objectivity generated by the use of agile development. Software projects only succeed with effective planning, visibility, and coordination. Agile practices promote a disciplined project management.

  My experience with agile development was great, in my opinion it is essential to use this methodology because it makes the development process more objective and clearer. (Participant B)

An important aspect to be observed in the use of agile practices. Thus, Figure 1 presents agile practices used by the participants to develop mobile applications during the course.

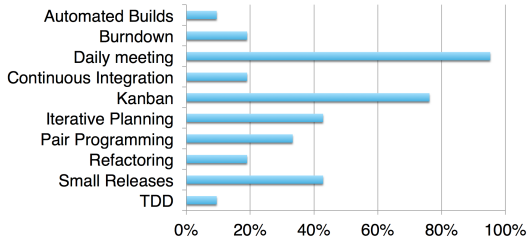


Figure 1: Agile practices used by participants in this study.

The majority of the participants adopt daily scrum meeting practice, because it helps them to keep track of project activities and communication. The second practice more adopted by the participants is Kanban (a system for visualising work do be done, in progress or completed). By the adoption of Kanban participants can see the progress of each activity, what still need to be done, what is in progress and what is completed. Iterative planning was also reported by participant. This practice help to organize the development and deliverables on different interactions and continues improvement. Small releases is also used by participants in order to organize different deliverables in accordance to the iterative planning. Pair programming was also reported as very useful specially when participants need to learn something new or need to work on something critical. Burndown, continuous integration and refactoring were reported by less than 20% of participants. Automated builds and TDD (Test Driven Development) were reported by less than 10% of participants.

6 DISCUSSION

Developing mobile applications can be hard due to many reasons. In this study, we found five main challenges. These challenges are faced for both beginners practitioners as well as more experienced developers. We also identified eight benefits of the adoption of agile practices for mobile application development.

The majority of the answers given by interviewees (50%) reported UI/UX as challenge for mobile application development. According to Dalmasso et al. (Dalmasso et al., 2013), most of the developers would like to release apps for major mobile platforms (iOS,

Android) and provide a consistent UI and UX across the platforms. However, developing an app for separate mobile platforms require in-depth knowledge of their SDKs (Software Development Kit). The developer can control all aspects of the user experience, but a mobile application must share common elements of the user interface with other applications and must adhere to externally developed user interface guidelines (Wasserman, 2010). The diversity of mobile platforms, as well as the variety of SDKs and other tools contributes to increase this challenge.

Different users' expectations and different development platforms are reported in 30% and 20% of the answers, respectively. This result shows that the main elements of mobile applications, user and technology, can pose challenges in mobile development. As well as, it poses challenges in teaching and learning mobile software development. We believe that this challenge will increase over the years. It can happen due to the increase number of new users and technologies. A single mobile application can reach millions of users with different devices, age groups, and supported by different platforms.

Continuous update and devices applications performance are reported in 10% of the answers given by interviewees. These challenges have a lower percentage when compared to define UI/UX challenge. However, these challenges are not less important, and in fact mobile applications are becoming more complex and users require high-quality mobile apps (Wasserman, 2010).

We also identified the benefits of the adoption of agile practices for mobile application development. The greatest benefit according to our findings is to improve the management and control. It makes sense, since agile approaches are focused project management (Scharff and Verma, 2010). At the same time, agile practices help to increase the development speed. It is very important in the mobile market since new applications are available every day in the Apps store.

The benefits of agile practices adoption described in this study are not necessarily restricted to the specific type of software development and it can also be extended to other software application domains. On the other hand, we identified challenges in mobile application development domain. A further investigation should be conducted in order to explore the relationship between challenges and achieved benefits.

An unexpected benefit from the adoption of agile practices was presented in terms of students engagement and motivation. We did not report this benefit in Section 5.2. However, it is important to highlight its contribution for teaching and learning mobile application development.

Our study is helpful in uncovering the underlying challenges and their implications on existing practice. First, challenges identified enable further research on more detailed activities important to consider while implementing an agile project for mobile application domain. Second, the benefits reported here have been mentioned by the interviewees and identified during the coding process alongside the challenges. Similarly to the challenges, some benefits can be more perceived than others.

6.1 Limitations of this Study

Our study was conducted with a limited number of respondents and from the same iOS development course. In addition, our results are drawn the viewpoint of students (development teams). It is also important to notice that part of project participants were attending a training course without previous experience with other approaches or software practices. These features highlight the fact that participants may become comfortable with it, and accepted the environment challenges and its limitations.

However, our results demonstrated that on using agile practices as part of a mobile application development environment are similar to previous literature studies. Our results have also shown that short development cycles and small releases are important features on mobile application development environments. We have found indicatives in our study that agile practices are the best approaches for mobile software development environments.

7 FINAL REMARKS

This study explores the adoption of agile practices for mobile application development. In other words, we investigate challenges and the students' experience on the adoption of agile practices. We identified five main challenges in mobile application development and eight benefits of agile practices for developing mobile applications.

Our results show that the main challenge to develop mobile applications is to define UI and UX followed by achieve different users' expectations. Regarding to the benefits, we found improvements on management and control as well as development speed. All teams finished their application projects (apps) delivering more than five different applications covering areas such as games, public transportation, services and productivity. Their apps presented a high

quality and used advanced resources such as data persistence, web services, etc.

Results from our study can be used to support developers, project managers, decision makers, and practitioners in order to choose the software development methodology to develop a mobile application project.

For future work, we will use the findings of this study to design an approach for teaching and learning mobile application development. The adoption of agile practices for mobile application development will be further investigate in order to propose new practices and processes to support software development.

ACKNOWLEDGMENT

Afonso Sales is funded by CNPq-Brazil (Universal 470096/2013-6) and Paulo Fernandes is also funded by CNPq-Brazil (PQ 307602/2013-3).

REFERENCES

Alston, P. (2012). Teaching Mobile Web Application Development: Challenges Faced and Lessons Learned. In *Proceedings of the 13th Annual Conference on Information Technology Education*, SIGITE '12, pages 239–244, Calgary, Alberta, Canada. ACM.

Beck, K. (2000). *Extreme Programming Explained Embrace Change*. Addison-Wesley, USA, 1st edition.

Corral, L. (2012). Using Software Quality Standards to Assure the Quality of the Mobile Software Product. In *Proc. of the 3rd Annual Conference on Systems, Programming, and Applications: Software for Humanity (SPLASH'12)*, pages 37–40, Tucson, AZ, USA. ACM.

Dalmasso, I., Datta, S., Bonnet, C., and Nikaein, N. (2013). Survey, comparison and evaluation of cross platform mobile application development tools. In *Wireless Communications and Mobile Computing Conference (IWCMC), 2013 9th International*, pages 323–328.

Flora, H. K. and Chande, S. V. (2013). A review and analysis on mobile application development processes using agile methodologies. *International Journal of Research in Computer Science*, 3(4):9 – 18.

Huy, N. P. and vanThanh, D. (2012). Evaluation of Mobile App Paradigms. In *Proceedings of the 10th International Conference on Advances in Mobile Computing and Multimedia*, MoMM '12, pages 25–30, Bali, Indonesia. ACM.

Kaleel, S. B. and Harishankar, S. (2013). Applying agile methodology in mobile software engineering: Android application development and its challenges. Technical report, Department of Computer Science, Ryerson University.

Lewis, G. A., Nagappan, N., Gray, J., Rosenblum, D., Muccini, H., and Shihab, E. (2013). Report of the 2013 ICSE 1st International Workshop on Engineering Mobile-enabled Systems (MOBS 2013): 12. *SIGSOFT Software Engineering Notes*, 38(5):55–58.

Macias, F., Holcombe, M., and Gheorghe, M. (2003). A Formal Experiment Comparing Extreme Programming with Traditional Software Construction. In *Proceedings of the Fourth Mexican International Conference on Computer Science*, pages 73–80.

Moore, A. and Flannery, W. (2007). Use of Extreme Programming Methodologies in IT Application Design Processes: An Empirical Analysis. In *Portland International Center for Management of Engineering and Technology - Management of Converging Technologies*, pages 2468–2475, Portland, OR, USA.

Paasivaara, M., Durasiewicz, S., and Lassenius, C. (2008). Distributed agile development: Using scrum in a large project. In *Global Software Engineering, 2008. ICGSE 2008. IEEE International Conference on*, pages 87–95.

Reichlmayr, T. (2011). Working towards the student Scrum - Developing Agile Android applications. *ASEE Annual Conference and Exposition, Conference Proceedings*.

Santos, A., Sales, A., Fernandes, P., and Nichols, M. (2015). Combining Challenge-Based Learning and Scrum Framework for Mobile Application Development. In *Proc. of the 2015 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE'15)*, pages 189–194, Vilnius, Lithuania.

Scharff, C. and Verma, R. (2010). Scrum to Support Mobile Application Development Projects in a Just-in-time Learning Context. *Proceedings - International Conference on Software Engineering*, pages 25–31.

Smite, D., Moe, N. B., and Gerfalk, P. J. (2010). *Agility Across Time and Space: Implementing Agile Methods in Global Software Projects*. Springer Publishing Company, Incorporated, 1st edition.

Wasserman, A. I. (2010). Software Engineering Issues for Mobile Application Development. In *Proceedings of the FSE/SDP Workshop on Future of Software Engineering Research*, FoSER '10, pages 397–400, Santa Fe, New Mexico, USA. ACM.

Xanthopoulos, S. and Xinogalos, S. (2013). A Comparative Analysis of Cross-platform Development Approaches for Mobile Applications. In *Proceedings of the 6th Balkan Conference in Informatics*, BCI '13, pages 213–220, Thessaloniki, Greece. ACM.

Zhang, Y. (2015). Development of Mobile Application for Higher Education: An Introduction. In Zhang, Y. A., editor, *Handbook of Mobile Teaching and Learning*, pages 1–4. Springer Berlin Heidelberg.

Knowledge Fusion for Cooperative Innovation from Strategic Alliances Perspective

Jucheng Xiong and Li Li
Shenzhen Graduate School, Harbin Institute of Technology, 518055, Shenzhen, China
xjc1252000@163.com, ximlli@126.com

Keywords: Knowledge Fusion, Cooperative Innovation, Strategic Alliances, IT-based Knowledge Management.

Abstract: It has been argued that strategic alliances offer opportunities for using purposive inflows and outflows of knowledge to accelerate cooperative innovation. In this introductory article, we seek to identify the means by which knowledge fusion helps create new knowledge and technological innovations. By analyzing the previous researches in terms of fusion and collaboration, we summarize the approaches of knowledge fusion based on IT application. Meanwhile, we also give effectiveness mechanisms and brief agenda for research in this important area. This study offers deep theoretical and managerial insights for firms and other institutions to manage knowledge fusion in strategic alliances.

1 INTRODUCTION

A growing trend in today's innovation environment is intensification of co-competition. In order to compete in a global market, more and more distributed organizations bound to work in alliances to gather and share knowledge by using information technology. Currently enterprises often establish strategic alliances such as patent pools, industry-university collaborative innovation alliances, and industrial technology innovation alliances to cocreate value that involves the sharing of knowledge and expertise for developing new or better products (Dyer and Hatch, 2006; Grover and Kohli, 2012). As noted by Grant (1996), knowledge is the preeminent resource of the firm and organizational capability involves integration of distributed knowledge bases. To maximize the benefits of knowledge integration emanated in multiple organizations environment, the issue of knowledge fusion and innovation gained through collaboration is important (Meijer, 2000; Rundquist, 2014).

Knowledge fusion is defined as recognition and combination of knowledge that are located and extracted from multiple, distributed, heterogeneous sources to generate new products, services, processes, capabilities or competencies (Preece et al., 2001; Heffner and Sharif, 2008). Most contemporary organizations are pursuing competitive advantage from the management information systems. Advanced information technologies (e.g., the Internet, Word Wide Web, distributed information systems, data mining and searching, simulation and modelling) can enhance the ability to recognize, assimilate, and exploit external knowledge (Alavi and Leidner, 2001; Dittrich and Duysters, 2007). However, most research on knowledge fusion are focusing on IT level (e.g., the ontology, fusion framework, fusion algorithm, multi-agent systems), while this is not enough for knowledge fusion, with many problems remaining to be solved from the knowledge management perspective.

There is a growing stream of literature investigating inter-organizations knowledge management in innovation alliances (Christoffersen, 2013; Vasudeva et al., 2013; Li et al., 2014), in which collaborators and competitors integrate in the pursuit of the codevelopment of technological innovations (Han et al., 2012). Knowledge fusion has been studied as a conversion procedure in knowledge integration with a focus on IT tools to support knowledge availability, sharing, and assimilation. In this paper we take one step toward addressing the gap between engineering science and knowledge science in prior research. We seek answers to the following set of questions for knowledge fusion management: What conditions facilitate knowledge fusion in innovation alliances? What management mechanisms are the most effective in enabling knowledge fusion?

Xiong, J. and Li, L.
Knowledge Fusion for Cooperative Innovation from Strategic Alliances Perspective.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 498-503
ISBN: 978-989-758-187-8

2 CONTEMPORARY RESEARCH THEMES

The knowledge-based view (KBV) suggest that superior profitability is likely to be associated with resource and capability-based advantages which are likely to drive from superior access to and integration of specialized knowledge (Grant, 1996). In order to support the knowledge integration, much of the research into the management issues concerning the role of information technologies has been focusing on the knowledge management system (KMS) (Černe et al., 2013; Sutanto and Jiang, 2013; Wang et al., 2014). Contemporary environment with open information systems (Li et al., 2014) make the combinative capabilities become more and more important. This ability of the firm to generate new combinations of existing knowledge is improved with the knowledge fusion theory developed.

The academic results and practical applications of KRAFT (Knowledge Reuse and Fusion/ Transformation) project are considered the most representative study in knowledge fusion research. KRAFT is conceived to investigate how existing proposals for distributed information systems architectures can support fusion of knowledge in the form of constraints expressed against an object data model (Gray et al., 1997). The literature on knowledge fusion in the field of computer science has explored the role of KMS in knowledge storage, sharing, reuse, revealing, generation, entry, integration, transportation, search and indexing (Preece et al., 2001; Smimov et al., 2013). The primary emphasis of this literature is on the architectures and fusion algorithms (Jiang et al., 2012; Zhou et al., 2013).

At the same time, research in knowledge fusion among multiple organizations has raised several questions that must be addressed. Heffner et al. (2008) articulate the knowledge fusion for technological innovation in organizations as a critical theme for future research. They propose that we need to integrate a number of heretofore disparate research streams, thereby providing a management framework for examining the knowledge fusion activities of organizations connect current researching on knowledge management. A management attitude towards knowledge fusion and innovation is discussed by Meijer (2000), who points out that problem solving comes down to creative processes which very much depend on thought processes that primarily take place inside the brains of individuals, under the influence of the group or the environment in which they do their creative work. By emphasizing how IT-based knowledge fusion is occurred in innovation alliances, fusion mechanisms research can help decision making and problem solving. Figure 1 illustrates the knowledge fusion management framework in strategic alliances.

3 KNOWLEDGE FUSION IN INNOVATION ALLIANCES

The capacity of the information technology to capture, store, and analyze information offers many opportunities for cocreation of business value (Grover et al., 2012), especially in alliances that trust and formal contracts can offer opportunities for knowledge sharing and leveraging. Traditionally, innovation has been created and marketed under closed settings, in which companies internally manage all of the processes involved in the innovation life cycle. Despite the nascent stage of development, many contemporary business enterprises have jumped on the bandwagon of the emerging industrial trend, participating in open

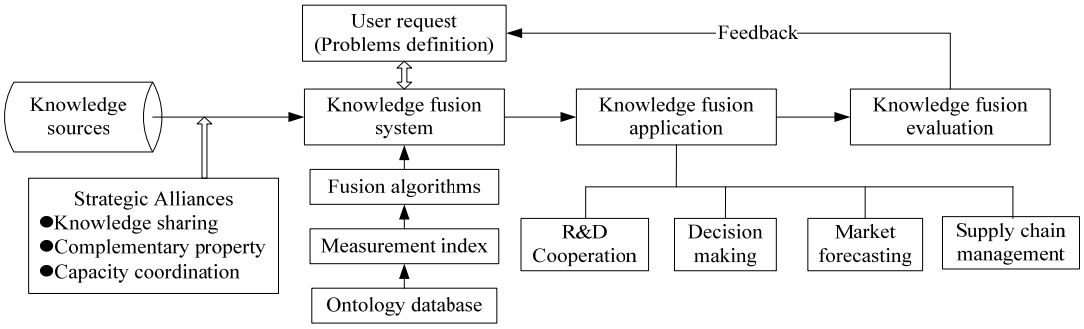


Figure 1: Knowledge fusion management in strategic alliance.

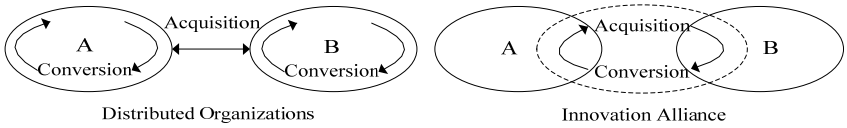


Figure 2: Two knowledge fusion approaches in different environment.

innovation alliances in pursuit of leveraging purposive knowledge inflows and outflows (Han et al., 2012).Knowledge acquisition and conversion are crucial to the knowledge fusion, since it makes possible a much greater degree of innovation ability. Figure 2 illustrates the two knowledge fusion approaches through which firms can acquire and convert knowledge in different ways.

Firms are usually to exploit external knowledge sourcing by capturing or engaging in alliances. From transaction cost economics (TCE) perspective, capturing takes place when organizational boundaries exist and knowledge is valuable, which influenced by the risk of opportunism, information asymmetries, and asset specificity. In contrast, the resource based view (RBV) can extend understanding of firm boundaries because it explicitly recognizes knowledge as a critical resource (Carayannopoulos and Auster, 2010). Joint and interactive learning represents a coupled form of knowledge fusion (Rosell et al., 2014), where acquisition and conversion take place through cooperative efforts between organizations that maintain their separate identities while sharing inputs and control. It seems that innovation alliances offer an interesting context within which knowledge fusion can be studied. Knowledge fusion can also be facilitated by the prosperity of collaborators as well as rivals in multi-organizational environment.

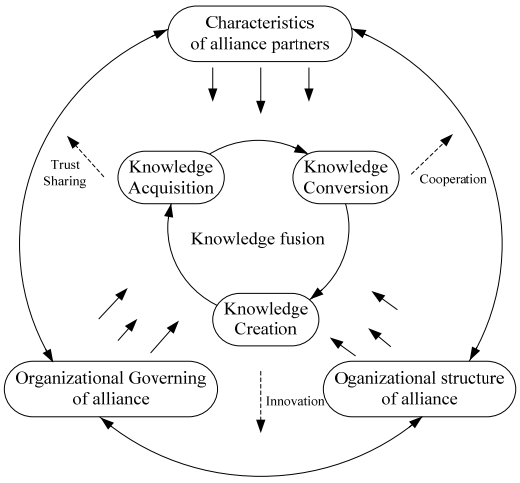


Figure 3: Innovation alliance features effect on knowledge fusion.

Many studies have examined the role of knowledge management in alliances (Mesquita et al.,2008; Shin and Lee, 2013). However, we limit our review to the studies that focus on knowledge fusion between two alliance partners and its impacts on the cocreating innovation capabilities. Fusions refers generally to the blending of different things into something new, something more than the mere sum of the parts, which in the process of combination release or generate tremendous energy . Based on an analysis of KRAFT project, the core of knowledge fusion is the knowledge conversion which depends on an iterative exploration cycle and information application by capturing both explicit and tacit knowledge. The first step of knowledge fusion is knowledge acquisition which established on the basis of the knowledge sharing. In innovation alliances, it's facilitated to acquire external knowledge sourcing through cooperation between organizations that maintain their separate identities while sharing complementary capability and assets. Figure 3 illustrates the innovation alliance features effect on knowledge fusion.

4 EFFECTIVENESS MECHANISMS FOR KNOWLEDGE FUSION

We believe that IT-based knowledge fusion from distributed databases and knowledge bases represents one of the most important streams in creativity and innovation that will gain greater importance as firms expand collaborative relationships in innovation alliances. In order to strengthen and promote knowledge fusion we offer some brief effectiveness mechanisms to solve the problems and challenges in practice (Table 1).

1. To expand the knowledge source network. Our framing drew largely from the strategy alliance perspective with the assumption that firms will form a cooperative bond and be willing and able to share knowledge through thoughtful use of IT. However, there are several other aspects that need to be emphasized in order to set a comprehensive research agenda.

Table 1: Effectiveness mechanisms for knowledge fusion.

	Prerequisites	Enablers	Results
Knowledge Source Network	Knowledge sharing and leveraging	● Position ● Size ● Density	Knowledge spread
Knowledge Fusion Process	Knowledge acquisition, conversion, creation, and application	● Relationship ● Complementary ● Synergic	Dynamic and continuous set of processes and practices
IT-based Knowledge Fusion Support System	IT infrastructures and elegant thinking	● Brain ● Machine intelligence	System quality, information quality, and usefulness
Management Initiatives	The role and impact of IT	● Organization ● Control ● Feedback	Virtuous circle of knowledge fusion and innovation

For instance, although innovation alliances offer opportunities for knowledge sharing and leveraging beyond the firm boundary, they also carry the risk of knowledge leakage to partner firms. Furthermore, of the two main types of knowledge, explicit and tacit, the latter is especially important due to its limited transferability because the tacit knowledge is acquired by and stored within individuals in highly specialized form. In order to solve these problems, it is necessary to expand the knowledge source network, which should not only focus on the knowledge bases but also build some efficient communication channels (e.g., expert systems, discussion forums, knowledge directories, and public innovation platform). For this proposes it is reasonable to reduce potential barriers in knowledge sharing between firms, explore the tacit knowledge transfer ways and means, increase intelligibility of knowledge representation for the users, and promote the spread of open knowledge sources.

2. To focus on the knowledge fusion process. While conceptually the idea of knowledge fusion is intuitive and simple, the process through which innovators can successfully implement it is likely to pose several challenges. How locate data and knowledge relevant to their current needs. The ability of knowledge acquisition which involves searching and retrieving from a wide array of knowledge is the prime condition. This process decides the quantity and quality of the available knowledge resources for knowledge conversion and creation. Regarding interdependencies, the ultimate goal of the knowledge fusion is to use the new knowledge in practice. One of the important implications of the framework is that knowledge fusion consists of a dynamic and continuous set of processes and practices embedded in individuals, as well as in groups and IT structures. So the process of knowledge fusion is not discrete and independent. Another implication of this framework is that knowledge fusion processes of acquisition, conversion, creation, and application are essential to effective innovation. We contend that the application of IT can create an infrastructure and environment that contribute to knowledge fusion by actualizing, supporting, augmenting, and reinforcing the fusion processes.

3. To develop IT-based knowledge fusion support system. The knowledge fusion support systems heavily rely upon advanced IT infrastructures. Our analysis of the literature suggests that IT can lead to a great depth and breadth of knowledge fusion in organizations. Usually, the knowledge fusion system architecture includes the construction of meta-knowledge, calculation of fusion knowledge metric, knowledge fusion algorithm, and post processing for fusion knowledge, all of these function modules are depend on the IT tools and capabilities. As with most information systems, the success of knowledge fusion support system partially depends upon the extent of use, which itself may be tied to system quality, information quality, and usefulness. At the current stage the knowledge fusion patterns and algorithms are hot research topic in some specific area, but they are not enough to support the common knowledge fusion systems. Some future research is needed such as agent architectures, prototypes for knowledge sharing, virtual reality-based ontology, algorithms and cooperation models. Thus, building IT-based knowledge fusion support system needs comprehensive consideration of knowledge management and information systems.

4. To find the relationship between IT and knowledge fusion management initiatives. It is important to note that managing knowledge fusion in innovation alliances is an important issue and that the main challenge is primarily related to the role and impact of IT. We have discussed the potential role of IT relates to more extensive network and

communication channels, faster access to knowledge, just in time learning, and more rapid application of new knowledge. Meanwhile, we should clear that the actual knowledge fusion for problem solving only happens in the minds of humans. It is the manager's task to provide the technical and environment in which the innovators are inspired to be creative and feel free to communicate. Managers should realize that IT tools are used to support the human's creative work but the IT-based systems themselves are incapable of keeping pace with dynamic needs of knowledge fusion. So the most important consideration is to coordinate machine intelligence and human creativity when individuals or teams engage in a cooperative research and development project. This could create a virtuous circle of knowledge fusion and innovation.

5 CONCLUSIONS

In this paper, we have presented a discussion of knowledge fusion in innovation alliance based on a review, interpretation, and synthesis of a broad range of relevant literature. We also have highlighted IT-based knowledge fusion that is of increasing importance for firms that seek to be cooperative and innovative. With respect to innovation, innovators can be involved in multiple knowledge fusion process chains. In order to solve problems and make decisions, knowledge fusion can take place in human brains and intelligent machines with the help of IT. The patterns and algorithms are the core modules in the knowledge fusion model. Furthermore, we have given effectiveness mechanisms from four layers: knowledge source network, the process of knowledge fusion, IT-based knowledge fusion support system, and management initiatives.

Through this special issue, our goal is to seek effective ways to manage the IT-based knowledge fusion for innovation. As we summarize above, an outline of the knowledge fusion system have been described from the co-competitive perspective. The analysis also yields some conclusions that are potentially important for firm managers and alliance practitioners. They need to regard the choice of knowledge disclosure level and reduce the transaction costs in the process of knowledge acquisition. As the information technology entered a big data era, dynamics of competition and cooperation among firms continue to evolve, and IT-based infrastructures, devices, and software tools create opportunities for knowledge fusion. The ongoing work includes available knowledge resources, advanced man-machine interactive, efficient knowledge fusion patterns and algorithms, consistent update knowledge database, and effective new knowledge evaluation.

ACKNOWLEDGEMENTS

Research works in this paper are financially supported by Soft Science Research Project of Guangdong (Grant No. 2013B070206002), Research Planning Foundation in Humanities and Social Sciences of the Ministry of Education of China (Grant No. 13YJAZH044) and National Natural Science Foundation of China (Grant No. 61173052, and No. 71103050).

REFERENCES

Alavi, M., and Leidner, D. E. 2001. Review: Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues. *MIS Quarterly*, 25(1): 107-136.

Carayannopoulos, S., and Auster, E. R. 2010. External Knowledge Sourcing in Biotechnology through Acquisition Versus Alliance: A KBV Approach. *Research Policy*, 39(2): 254-267.

Černe, M., Jaklic, M., and Skerlavaj, M. 2013. Management Innovation in Focus: The Role of Knowledge Exchange, Organizational Size, and IT System Development and Utilization. *European Management Review*, 10(3): 153-166.

Christoffersen, J. 2013. Cooperation in International Strategic Alliances and Impact on Host Economies: Knowledge Transfer and Diffusion to Local Firms. *European Journal of Development Research*, 25(4): 518-536.

Dittrich, K., and Duysters, G. 2007. Networking as a Means to Strategy Change: The Case of Open Innovation in Mobile Telephony. *Journal of Product Innovation Management*, 24(6): 510-521.

Dyer, J. H., and Hatch, N. W. 2006. Relation-Specific Capabilities and Barriers to Knowledge Transfers: Creating Advantage through Network Relationships. *Strategic Management Journal*, 27(8): 701-719.

Grant, R. M. 1996. Prospering in Dynamically-Competitive Environments: Organizational Capability as Knowledge Integration. *Organization Science*, 7(4): 375-387.

Gray, P. M. D., Preece, A., and Fiddian, N. J. 1997. KRAFT: Knowledge Fusion From Distributed Databases and Knowledge Bases. *Proceedings. Eighth International Workshop on Database and Expert Systems Applications* (Cat. No. 97TB100181): 682-691.

Grover, V., and Kohli, R. 2012. Cocreating It Value: New Capabilities and Metrics for Multifirm Environments. *MIS Quarterly*, 36(1): 225-232.

Han, K., Oh, W., and Im, K. S. 2012. Value Cocreation and Wealth Spillover in Open Innovation Alliances. *MIS Quarterly*, 36(1): 291-315.

Heffner, M., and Sharif, N. 2008. Knowledge Fusion for Technological Innovation in Organizations. *Journal of Knowledge Management*, 12(2): 79-93.

Jiang L, Liang K, Ye S. 2012. Knowledge Fusion Model Research Based On Granular Computing Theory. *Application Research of Computers*, 29(10):3697-3700.

Li, L., Sun, L., and Xu, L. 2014. Knowledge Fusion System for Open Innovation Alliances. *ICIC Express Letters*, 8(4): 1089-1096.

Meijer, B. R. 2000. A Management Attitude Towards Knowledge Fusion and Innovation. *Proceedings of the 2000 IEEE Engineering Management Society*. EMS - 2000 (Cat. No.00CH37139): 642-647.

Mesquita, L. F., Anand, J., and Brush, T. H. 2008. Comparing the Resource-Based and Relational Views: Knowledge Transfer and Spillover in Vertical Alliances. *Strategic Management Journal*, 29(9): 913-941.

Preece, A., Hui, K., and Gray, A. 2001. KRAFT: An Agent Architecture for Knowledge Fusion. *International Journal of Cooperative Information Systems*, 10(1-2): 171-195.

Rosell, D. T., Lakemond, N., and Wasti, S. N. 2014. Integrating Knowledge with Suppliers at the RandD-Manufacturing Interface. *Journal of Manufacturing Technology Management*, 25(2): 240-257.

Rundquist, J. 2014. Knowledge Integration in Distributed Product Development. *International Journal of Innovation Science*, 6(1): 19-28.

Shin, H., and Lee, H. 2013. Disentangling the Role of Knowledge Similarity On the Choice of Alliance Structure. *Journal of Engineering and Technology Management*, 30(4): 350-362.

Smirnov, A., Levashova, T., and Shilov, N. 2013. Patterns for Context-Based Knowledge Fusion in Decision Support Systems. *Information Fusion* (http://dx.doi.org/10.1016/j.inffus.2013.10.010).

Sutanto, J., and Jiang, Q. 2013. Knowledge Seekers' and Contributors' Reactions to Recommendation Mechanisms in Knowledge Management Systems. *Information ang Management*, 50(5): 258-263.

Vasudeva, G., Spencer, J. W., and Teegen, H. J. 2013. Bringing the Institutional Context Back in: A Cross-National Comparison of Alliance Partner Selection and Knowledge Acquisition. *Organization Science*, 24(2): 319-338.

Wang S, Noe R A, Wang Z. 2014. Motivating Knowledge Sharing in Knowledge Management Systems a Quasi-Field Experiment. *Journal of Management*, 40(4):978-1009.

Zhou, F., Wang, P., and Han, L. 2013. Multi-Source Knowledge Fusion Algorithm. *Journal of Beijing University of Aeronautics and Astronautics*, 39(1): 109-114.

Application of Metrics for Risk Management in Environment of Multiple Software Development Projects

Júlio Menezes Jr[1,2], Miguel Wanderley[1,2], Cristine Gusmão[1,3] and Hermano Moura[2]
[1]*SABER Tecnologias Educacionais e Sociais Research Group, Federal University of Pernambuco, Av. dos Reitores, s/n, Cidade Universitária, Recife, Pernambuco, Brazil*
[2]*Centre of Informatics, Federal University of Pernambuco, Av. Jornalista Anibal Fernandes, s/n, Cidade Universitária, Recife, Pernambuco, Brazil*
[3]*Department of Biomedical Engineering, Centre of Technologies and Geociences, Federal University of Pernambuco, Av. da Arquitetura, s/n, Recife, Pernambuco, Brazil*
{jvmj, mdsw}@cin.ufpe.br, cristine.gusmao@pq.cnpq.br, hermano@cin.ufpe.br

Keywords: Risk Management, Multiple Software Project Management.

Abstract: Multiple Project Management currently is a reality in software development environments. In the case of software projects, some characteristics are highlighted, such as constant changes in levels of scope or product, software complexity and aspects related to human resources, such as technical knowledge and experience, among others. We may consider these characteristics as risk factors that should be managed. In this aspect, a tactical management requires the usage of better-structured information, which leads us to think about the usage of a metrics-based strategy as a support tool for multiple project managers with emphasis on risk factors. In this context, this work presents an an application of the metric "Risk Points" and its variations in an environment of multiple software development project. This experience report aims to evaluate the proposed metrics as a decision-support tool and monitoring of risk during project life-cycle.

1 INTRODUCTION

Nowadays there is a consensus that, in software engineering, if adverse factors are not well managed, projects might fail. According to (The Standish Group, 2013) only 39% of software projects are completed on time and on budget. It is interesting to notice that the most of causes of project fail occur due to not managed risk factors. On the other hand, we realize that risk management in software engineering needs more practical and deep studies (Bannerman, 2014), allowing more concise identification of its practices as well as improvement points.

Despite the recognized importance, in practice the explicit risk management in software engineering is still limited. One of the reasons for this scenario is that risk is subjective in software projects. In this light, one way to reduce the subjectivity bias is using metrics, because it could be helpful to provide to the stakeholders a better knowledge, control and improvement of risk management processes adopted on environment of multiple software projects. Also, there is a clear gap about risk measurement in software engineering (Menezes Jr et. al., 2013).

One of the related works presents a proposal of metrics called "Risk Points" (Lopes, 2005), whose object is to measure the risk level of a project in an environment of multiple software development projects. The central idea is to help managers in decision-making for risk reduction, as well as to analyze the effectiveness of actions to do that.

Therefore, this paper presents a pilot experience of the Risk Point metrics application in a real environment of software development. The main goal is to evaluate the metrics and its effectiveness in an environment of multiple software projects.

After this introductory section, the rest of this paper is organized as follows: Section 2 brings and briefly discusses some related works; Section 3 introduces the proposed metrics and it alternatives; Sections 4 and 5 presents the experience report objectives and methodology, respectively; Section 6 shows the results of the presented methodology; the next section discusses these results. Finally, Section 7 presents final considerations and future work.

2 RELATED WORK

There are few references in software engineering

Jr., J., Wanderley, M., Gusmão, C. and Moura, H.
Application of Metrics for Risk Management in Environment of Multiple Software Development Projects.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 504-511
ISBN: 978-989-758-187-8

about the usage of metrics for project risk management. Barry Boehm (Boehm, 1989) is considered a pioneer in the application of risk management in software engineering. He proposed a software risk management framework focused on risk analysis. The activity of risk analysis in his work is defined as Risk Exposure calculation, which is defined as the multiplication between Probability of Risk versus Loss or Impact of Risk. This analysis is only used for risk prioritization.

The work (Lopes, 2005) proposes a way of to measure the risk level of a project through a metrics called Risk Point. According to the author, the objective of Risk point metrics is to define how risky is a software project based on number of identified risks and project complexity factors. We use this metrics as one of the indicators for this dissertation. However, the author did not evaluate Risk Point in practice.

Another related work defines a quantitative approach where risk concepts of economics, specifically credit risk, are used to propose a method of risk assessment in software projects (Costa, 2005). In this work, the author proposes a way to calculate how much capital a software development organization can gain or lose due to the risks of a selected set of projects. The adopted method allows the selection of projects' sets that seeks to maximize the cost-benefit for an organization. The risk assessment method uses project characterization (size, duration cost and return) and a questionnaire to identify risks. However, this method was not evaluated in practice.

The use of the Goal-Question Metric paradigm to define software process metrics with the goal of monitoring risk factors is discussed on (Fontoura and Price, 2004). On the other hand, the proposal was not put in practice.

Some works used metrics for technical risks using Risk-Based Testing concept (RBT) (Amland, 2000) (Souza et al, 2009). The objective of the metrics is to indicate information regarding test cases control through risk analysis and monitoring of system requirements. However, these metrics are not proposed as a tool for management of projects, providing only product risk view based on system requirements, architecture and coding analysis.

Another related work discusses the need of the usage of metrics for risk management, and shows examples of how they can be used (Bechtold, 1997). For example, a risk factor related to team qualification – experience and knowledge level on certain technology. Hence, it is a data that could be quantified and followed through project life cycle. On the other hand, this paper does not present any practical application or assessment.

This paper approaches the evolution of the proposal presented by (Lopes, 2005) because it shows a proposal of a metrics – Risk point, whose goal is to measure risks in the context of multiple project software management as support tool for project managers. Therefore, the rest of this paper presents Risk Point metrics in details as well as proposes improvements and previous assessment in a real environment.

3 RISK POINT METRICS

The Risk Point (RP) metric aims to represent the overall risk exposure level of a project (Lopes, 2005). Basically, the metric is defined in terms of the amount of identified risks, where these risks are defined in terms of its probability and estimated impact, as the concept of Risk Exposure (RE) (Selby, 2007).

RP allows quantifying the project in terms of its identified risks. It is necessary to estimate the Risk Exposure value, i.e. Probability versus Impact, for each identified risk, so, for a specific data collection about the current risks of a project, it is possible to determine a value of Risk Point (RP), as follows:

$$Risk\ Point = PCF \times URPW$$

Where, **PCF** means the **Project Characteristics Factor** and **URPW** means **Unadjusted Risk Point Weight**. PCF is a value for giving the project a weight and adjust the metric final value based on technical and environmental factors (Coelho, 2003). This value is defined through the answers of a questionnaire, which was developed from an empirical study with software project managers and management students, as mentioned. Then, PCF is defined as:

$$PCF = 1.05 + (0.015 \times CF)$$

$$CF = \sum_{i=1}^{8} (Question_i \times Weight_i)$$

CF means **Characteristic Factor**, it is determined by answering the 8 questions of a questionnaire with scores between 0 and 4, and then this answer is multiplied by the defined weighted value for each question. Finally, these 8 products are summed, resulting in the CF value (Coelho, 2003).

URPW is the Unadjusted Risk Point Weight, composed by the identified risks during a data collection, in terms of their Risk Exposure. In this study, the estimation adopted was values in {0.1, 0.2, … , 0.9}.

The Unadjusted Risk Point Weight (URPW) value is formed by the summation of the Weights of each identified risk, being this Weight defined according the Risk Exposure value, as can be seen in the following table.

Table 1: Unadjusted Risk Point Weight (URPW) values.

Classification	RE(Risk)	Weight(Risk)
Very Low	[0.0, 0.2)	1
Low	[0.2, 0.4)	2
Average	[0.4, 0.6)	3
High	[0.6, 0.8)	4
Very High	[0.8, 1.0]	5

Thus, for n identified risks, the URPW value follows the rule:

$$URPW = \sum_{i=1}^{n} Weight(Risk_i) \mid n = number\ of\ identified\ risks$$

Briefly, a given data collection (even in a subjective way, with values in a 5 levels scale for Probability and Impact) about the current risks of a project yields a value which represents the overall evaluation concerning the known risks of a project in a specific moment in its life cycle. This value allows a broad risk assessment about the risk exposure level of a project in different moments, and also allows a way to compare between different projects based on their identified risks.

3.1 Alternative Metrics

Just changing the weights for the Risk Exposures classification, showed in Table 1, new alternative metrics were defined. Note that by changing the weights values we can create many other metrics, but the ones presented in this paper focus on the concept, taken as the most important, inside these changes.

Pure Risk Point (PRP). In this alternative metric, all the weights from URPW are defined as 1. Therefore, the URPW value composition becomes a simple summation of all identified risks, without distinguishing the different Risk Exposure values of each risk. PRP metrics prioritize the assessment of the number of different risks identified during some data collection.

Exponential Risk Point (ERP). This metric presents the weights from URPW in a base 2 exponential growth, i.e. {1,2,4,8,16}. Therefore, ERP is even higher for the highest occurrences of Risk Exposure levels. The URPW receives higher values for "Average" or upper levels of Risk Exposure. Therefore, this metrics is more sensitive for high risk exposures levels.

Criticality (CRIT). It is represented by the difference ERP – RP. Therefore, the difference is only visible when the risk exposure levels are defined as medium, high and very high. CRIT is defined as:

$$CRIT = \text{ERP} - RP$$

This metrics reveals the risks for high values, taking into consideration only the most critical risks in an assessment. Finally, for better understanding of the differences between the proposed metrics, the Table 2 presents the weights defined for each metrics.

Table 2: Weight values of each metrics.

Classification	RE (Risk)	W (RP)	W (PRP)	W (ERP)	W (CRIT)
Very low	[0.0, 0.2]	1	1	1	0
Low	[0.2, 0.4]	2	1	2	0
Medium	[0.4, 0.6]	3	1	4	1
High	[0.6, 0.8]	4	1	8	4
Very high	[0.8, 1.0]	5	1	16	11

W = Weight of the risk according do Risk Exposure (RE) calculation to URPW.

The main difference between the metrics is basically the weight given to each identified risk: RP uses a sequential scale; PRP basically counts the number of risks; ERP highlights the difference for high level of risks and, finally, CRIT only considers risks factors with medium or higher levels.

Adjusted Metrics. It is possible to observe projects with different number of risks in the same environment. To allow comparison between projects, in this work we divided the metrics by the number of identified risks:

$$Adjusted\ Metrics = Metrics\ /\ Number\ of\ identified\ risks$$

With this adjustment, it is possible to evaluate directly the values of the metrics, independently of the number of identified risks of each project.

4 OBJECTIVE

The main objective of this work is to evaluate the applicability of the proposed metrics and their

effectiveness in risk assessment in an environment of multiple software projects. To do so, each week information about risks were collected in five projects in the same environment. For each project, risk factors were identified and analyzed using predefined scales of probability and impact of each risk. Next sections present the methodology and results of the experience report.

5 METHODOLOGY

To execute the study, we used an agile risk management process called GARA (Ribeiro et al, 2009), consistent with agile development methodologies, such as Scrum, focused to multiple projects and simple enough for the risk management activities, such as the data collecting. The metrics were applied in a software development environment from a research laboratory at Federal University of *Pernambuco* (UFPE) specialized in educational technologies, in which weekly data collecting of information about risks were performed during 2 months. All the projects involve software applications on educational technologies.

Five projects were monitored between May 2015 and July 2015 together with their leaders. The projects are related to software development like web platforms – front and back-end, web services and mobile application. The following steps were executed:

1. **Risk Identification:** through a combination of brainstorming and the Risk Taxonomy from Software Engineering Institute (Carr et al, 1993). Additionally, project characterization factors were valued.
2. **Risk Assessment:** for each identified risks, values of probability and impact are calculated. For this work, we adopted the following values:

Table 3: Values of probability and impact used.

Name	Value
Very low	0.1
Low	0.3
Medium	0.5
High	0.7
Very high	0.9

3. **Data Processing:** with the raised information, the identified risks are categorized as from project and from environment. In this work project risks appear on only single project and environment risks appear on more than one project. With the collected information in the previous steps, the metrics calculation is made.
4. **Risk Controlling and Monitoring:** consists on the following-up of risk levels evolution of each project.

It is important to notice that the steps were performed weekly. Below we present some information about each project used in this study – description main product, number of participants and duration:

Project 1: web system to support to students' subscription in post-graduation and extension courses, including management of data and reports generation.

- Product: system information in web platform, front-end and back-end.
- Teams: software development (2) and design (3).
- Duration: 6 months

Project 2: system information for management of academic works, including term papers for undergraduate and graduate courses. This project has 3 important sub products: term paper elaboration and discussion forum, management reports and CRUDs requirements.

- Product: system information in web platform, front-end and back-end.
- Teams: software development (3) and design (3).
- Duration: 10 months.

Project 3: mobile system to access to educational contents about healthcare stored in external repositories. The system demands an external authentication server and the server side of the system is developed by another institution.

- Product: mobile application developed with Android platform.
- Teams: mobile development (4) and design (3).
- Duration: 12 months

Project 4: support-components for a distance course about primary healthcare, that includes virtual learning environment and a web portal.

- Product: web portal for access to the course, front-end, including visual and usability adjustments.
- Teams: web design (4), design (3) and virtual learning environment (1).
- Duration: 3 months.

Project 5: development of a system information, whose goal is to evaluate students present in educational platforms (Moodle) and management of them.

- Product: system information in web platform, front-end and back-end.
- Teams: web design (4), design (3) and virtual learning environment (1).
- Duration: 5 months.

6 RESULTS

During eight data collectings, the presented methodology were applied. Table 4 summarize results about number of identified risks.

Table 4: General results.

Total of identified risks	**31**
Total of Risk Exposure Mean	0.14
Number of identified risks – Project 1	30
Number of identified risks – Project 2	30
Number of identified risks – Project 3	22
Number of identified risks – Project 4	25
Number of identified risks – Project 5	26

Therefore, 31 different risks were identified in five projects. Considering the mean of Risk Exposure (Probability (risk) * Impact (risk)), most of the identified risks has low value. Table 5 presents the top ten risks from the environment, i.e, the ones with highest risk exposure value (average).

Its important to notice that the project leader is the responsible to valuate probability and impact according to information present on Table 1.

Table 5: Top ten risks.

Risk	**Average Risk Exposure**
Failures on deployment	0,25
Dependences of other teams	0.22
Dependence of specialists	0.22
Urgent demands, new demands raises	0.20
Conflicts with external activities of team members	0.20
Requirements changes	0.18
Team member absence	0.16
Team member unavailability	0.16
Exit of team member	0.16
Software testing process problems	016

For each project, all the identified risks (for respective risk exposure values) are synthetized in one single value. Therefore, the idea is to represent the overall risk level of each project in a specific moment. Figure 1, for example, presents the results of the application of Risk Points/Number of identified risks. X axis represents the number of weeks, whereas Y axis represents the metrics value.

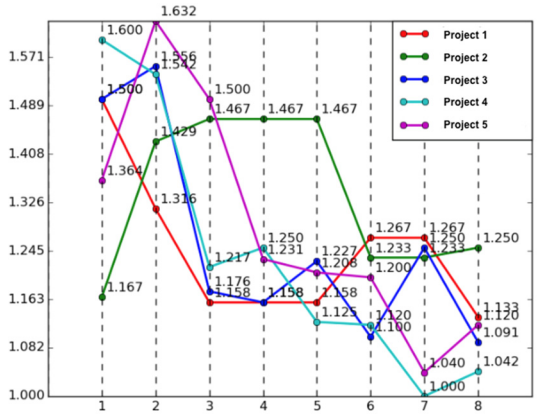


Figure 1: Risk Points / Number of risks.

Considering Risk Points metrics application, we can assume that, after 8 weeks, the Project 2 is the riskier one in the environment, whereas the Project 4 has presented a high level of decrease.

Figure 2 presents the application of the metrics Pure Risk Points (PRP). As mentioned before, this metrics just represents the number of identified risks of each project.

It is important to mention that the variation of values in Figure 2 does not necessarily mean that new risks arose or they were removed from the risk list. It just represents the risks in which the calculation of risk exposure was made.

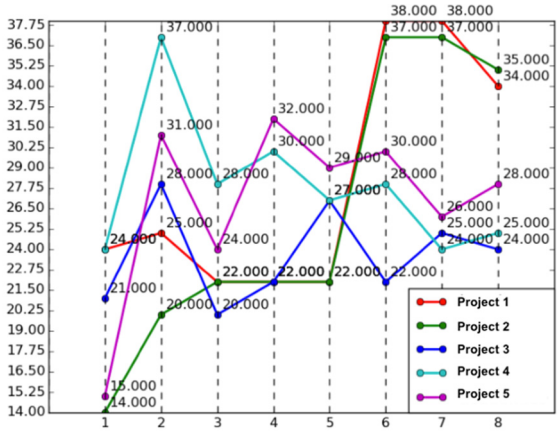


Figure 2: Pure Risk Points (PRP).

The application of Exponential Risk Point (ERP)/Number of identified risks is presented in the Figure 3.

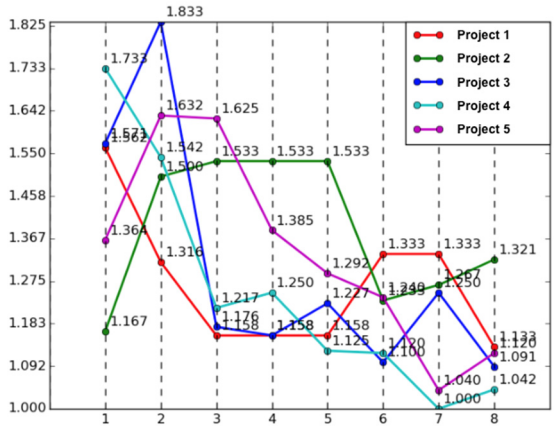


Figure 3: Exponential Risk Points (ERP) / Number of identified risks.

We can realize that the behavior of the Figure 3 is similar to the presented in the Figure 1. It happens because the differences between the metrics ERP e RP are noted only to the highest risk exposure value – between medium and very high.

The Project 1 was close to the end, so that it presented a considerable risk level in the last weeks. Before that, this project had a successful delivery.

The data collecting of the Project 2 started when it was beginning a new development cycle after an important milestone. At that moment, requirements have been risen and new demands have been grown. After the 5th week, the requirements were well defined, so that the risk level decreased. Even though this event, this project was considered as the riskier in the environment after eight weeks.

In the first month of data collecting, the Project 3 delivered an important release, that justifies the decrease of risk level during this period of time. In the second month, the team got test results with new requirements, adjustments and bugs to be fixed. It can explain the oscillation that happened in the second month of this project.

The Project 4 started as the riskier project and finished with the less risky one. The schedule of this project was relatively short, and it is similar to others that were finished and it was close to the end. In fact, this project was considered successful and did not present problems during its life-cycle.

Project 5 also was being finished. It presented a decrease during the period of assessment, just waiting for assessment, feedback and final approval of the testers.

In general, we realized that, after an important milestone, the risk level presents accentuate decrease. After the feedback, bugs identification and adjustments on the scope, the values start to grown and remains increasing until the next milestone or delivery of a release.

As explained before, to show the difference between RP and ERP, we used the metrics Criticality (CRIT). Figure 4 shows the moments in which the projects are presenting the most critical levels, i.e., with risk exposure level equal or greater than 0.5.

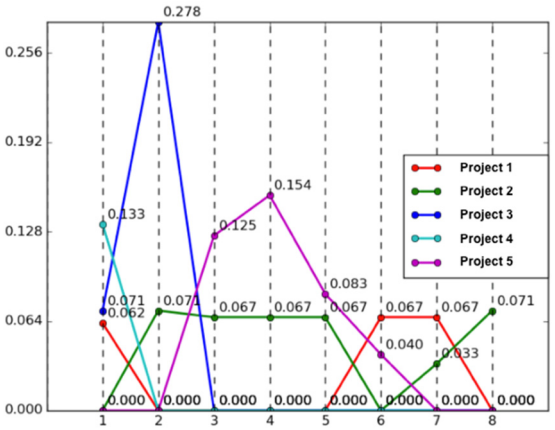


Figure 4: Criticality (CRIT) / Number of risks.

In the beginning of the study, the Project 4 presented the highest level of criticality and its value reduced to zero till the 8th week. In fact, this project had a relatively short schedule and it was finishing successfully, just waiting a final evaluation before the system deployment.

Project 2 was considered the most critical and riskier. It means that there were risks classified as medium or higher value. In the 8th week the project was close to an important release.

Project 5 was delivering a release and it was close to the end. The main functionalities were finished as it was agreed and it was just waiting a final feedback from a acceptance test. According to the leader of this project, the presented values were pertinent once it was really facing a critical phase between the third and sixth week.

Finally, the high value in the second week of the Project 3 was expected, once at that moment an important deliver was being finished. But the high difference between the others values needs a deeper investigation, because it can be a bias of the project leader.

7 DISCUSSIONS

An important characteristic identified in this work is: most of identified risks are classified as very low or

low. The impact of this in the metrics is the fact that the risks with high values are not well explored, even using ERP metrics. The low values of the metrics CRIT also shows this behavior.

Another point to be considered is that the processed values of the metrics presents two information: (i) it determines the risk level of a project with an only single value in a certain moment regarding the number identified risks; (ii) the experience and knowledge of each project leader and their respective skill to estimate the risks. Both information is crucial for a better comprehension of the context of the metrics application, because different people can perform different estimations in the same project. Therefore, the subjectivity bias still has to be taken into consideration, but the experience level of the project leader may be important to reduce it.

The risk list used was built using information given by project leaders during the first weeks. To guide the process of risk identification, we used the risk taxonomy of SEI (Carr et al, 1993), but only to make the brainstorming more focused. We did not used a predetermined risk list. Therefore, there is no evidence that the identified risks are the main ones of each project and from the environment. It also means that the metrics values are an estimation of the general level of project risk exposure.

8 CONCLUSIONS

This paper presented an experience report of the usage of the metrics Risk Points (Lopes, 2005) and proposed alternatives metrics in a real environment of software development projects. Next subsections bring main contributions, limitations and future opportunities of research.

8.1 Main Contributions

The main positive points of the proposed metrics show that they are capable to tell us, in only one single value, the general level of a software project risk exposure in a certain moment. Second, the metrics allows an assessment in environments of multiple projects, providing direct and indirect comparisons between different projects through their life-cycle.

The main negative point about the metrics is their sensitivity to experience level of the project manager and the accuracy level of them. In other words, the same project may have different values in the same moment when it is assessed by more than one person.

This work did weekly data gathering through online tools and meetings. At the end of the study, the project leaders made an assessment of the process and its effectiveness to improve knowledge about the projects risks. In general, the project leaders considered the study important for the process of project risk management.

In the first month of the study, all the data collection was face-to-face. This approach was efficient for the understanding of the projects, risk factors and, mainly, to make the process clearer. All the project leaders said that the presence of a risk manager is important to conduct risk identification and to make better estimations. However, this kind of meeting could be expensive, because demands more face-to-face meetings with approximately one hour of duration.

In the second month (last four gatherings), we applied an online questionnaire with the managers. The positive point of this approach was the flexibility and agility. However, we observed difficulties to assure that the project leaders answer the questionnaires on time.

One proposal for the process could be an intermediate approach, using both online and face-to-face in order to take advantage of the positive points of each one, using alternate iterations.

8.2 Future Work

Main directions for this work are to apply other case studies with some adjustments:

- Replicate the study for more projects during more time. Therefore, we can follow the behavior and identify noises and point of improvements;
- Analyze the main actions to mitigate levels of risks, taking it into consideration during the project life-cycle;
- Take into consideration the level of experience of the project manager/leader. It can be a value that may compose the metrics;
- Identify risk factors that are common in software projects. The idea is to work with a predetermined risk list to allow a better comparison between projects in the same environment;
- Perform research about the usage of knowledge base of risks, combining with the data collected, in order to predict risks for new projects.

ACKNOWLEDGEMENTS

The authors would like to thank *SABER Tecnologias Educacionais e Sociais* research group for the whole support to develop this work.

REFERENCES

Amland, S., 2000, Risk-based testing: Risk analysis fundamentals and metrics for software testing including a financial application case study, *J. Syst. Softw., vol. 53*, pp. 287–295.

Bannerman, P. L., 2014, A Reassessment of Risk Management in *Software Projects. v. 2*, p. 1119–1134.

Bechtold, R., 1997, Managing risks with metrics, *A term paper for MJY Team Software Risk Management WWW Site.*

Boehm, B.W., 1989, Software Risk Management, Lect. Notes Comput. *Sci., vol. 387*, p. pp 1–19.

Boehm, B., de Marco, T., 1997, Software risk management. *IEEE Software, v. 14, n. 3*, p. 17-19.

Carr, M., Konda, S., Monarch, I., Ulrich, F., Walker, C., 1993, Taxonomy Based Risk Identification. Software Engineering Institute, *Carnegie Mellon University*, USA.

Coelho, C., 2003, MAPS: Um Modelo de Adaptação de Processos de Software. *Master dissertation in Computer Science*. Universidade Federal de Pernambuco, Recife, Brazil.

Costa, H., 2005, Uma Abordagem Econômica Baseada em Riscos para Avaliação de uma Carteira de Projetos de Software. *Master dissertation. PESC/COPPE/UFRJ*, Rio de Janeiro, Brazil.

Fontoura, L., Price, R., 2004, Usando GQM para Gerenciar Riscos em Projetos de Software. *18º Simpósio Brasileiro de Engenharia de Software – SBQS P. 39 – 54.*

Freitas, B., Moura, H., 2004, GMP: Uma Ferramenta para a Gestão de Múltiplos Projetos. In: *Simpósio Brasileiro de Sistemas de Informação – SBSI.*

Gusmão, C., Moura, H., 2005, Gestão de Riscos para Ambientes de Múltiplos Projetos de Software: Teoria e Prática. *IV Escola Regional de Informática de Minas Gerais - IV ERI MG*, Belo Horizonte, Brazil.

Lopes, S., 2005, Análise e definição de métricas para o processo de gerência de riscos para projetos de software. *Graduation work. Centro de Informática*. Universidade Federal de Pernambuco. Recife. Brazil.

Menezes Jr., J.V., Gusmão, C. M. G., Moura, H. P., 2013, Defining Indicators for Risk Assessment in Software Development Projects. *CLEI Electronic Journal, v. 16*, p. 1-24.

Pressman, R., 2006, *Engenharia de Software. 6th edition*. São Paulo: McGraw-Hill.

Ribeiro, L., Gusmão, C., Feijo, W., Bezerra, V., 2009, A case study for the implementation of an agile risk management process in multiple projects environments, *Management of Engineering & Technology, 2009. PICMET 2009*. Portland International Conference, pp.1396,1404.

Selby, R.W., 2007, Software Engineering: Barry W. Boehm's Lifetime Contributions to Software Development, *Management, and Research*, John Wiley & Sons.

Souza, E., Gusmão, C., Alves, K, Venâncio, J., Melo, R., 2009, Measurement and control for risk-based test cases and activities. *10th Latin American Test Workshop, LATW*.

The Standish Group, 2013, *"Chaos Manifesto 2013."* Available at http://www.versionone.com/assets/img/files/CHAOSManifesto2013.pdf.

Wysocki, R., 2011, Effective Project Management: Traditional, Agile, Extreme. Wiley; 6 edition.

Linguistic Alerts in Information Filtering Systems
Towards Technical Implementations of Cognitive Semantics

Radoslaw P. Katarzyniak[1], Wojciech A. Lorkiewicz[1] and Ondrej Krejcar[2]
[1]*Faculty of Computer Science and Management, Wroclaw University of Technology, Wybrzeze Wyspianskiego 27, 50-370 Wroclaw, Poland*
[2]*Center for Basic and Applied Research, Faculty of Informatics and Management, University of Hradec Kralove, Rokitanskeho 62, 500 03 Hradec Kralove, Czech Republic*
{*radoslaw.katarzyniak, wojciech.lorkiewicz*}*@pwr.wroc.edu.pl, ondrej.krejcar@uhk.cz*

Keywords: Information Filtering, Linguistic Alerts, Computational Semiotics, Epistemic Modalities, Cognitive Semantics.

Abstract: An original model of natural language alerts production is proposed. The alerts are produced by information filtering system and stated in a quasi-natural language, both potentially written and vocalized. The alerts are chosen with respect to a certain collection of uncertain decision rules, thus they inherit various levels of epistemic uncertainty. The quasi-natural language statements include linguistic operators of epistemic modality, as their necessary parts. The proposed model implements in a technical context an adequate cognitive semantics captured by an original theory of epistemic modality grounding defined elsewhere.

1 INTRODUCTION

Users' selective dissemination of information and related information filtering (IF for short) are important challenges for modern information systems (Hanani et al., 2001). They seem particularly crucial for management executives, interested in and strongly dependent on up-to date information related to their everyday business activities (Xu et al., 2011). The way how the selected (filtered) information is presented needs to be designed with substantial influence of real environments in which the executives work, including these days frequent mobility of their daily work. In such circumstances all easily comprehensible presentation modes, for instance applications of quasi-natural, written and sometimes even vocalized languages, have become a very important theoretical and practical problem for computer science community.

Unfortunately in actual settings, it is often the case that on-line indexing of documents, incoming to executives' knowledge repositories, is practically impossible due to their inherent characteristics. For instance, a typical document can consists of expanded multimedia elements and therefore require advanced and time-consuming processing to elaborate semantic description of their content. Fortunately, at least in some practical contexts, an approximate (yet still effective) solution is to settle executive-oriented filtering solely on attributes of incoming documents for which values can be easily determined. Such attributes may include origin, author(s), affiliating institutions, attached general keywords, etc. However, a rather obvious inconvenience of the approximate solution is that filtering decisions may be uncertain to some extent. In particular, due to underlying soft classification rules in which preconditions are defined by means of easy-to-determine attributes, and post-conditions are built from subjects (topics) the executives are interested in. Another inconvenience might be that such IF systems need to be based on processes of classification rules management (namely, their effective extraction, storage, retrieval and update).

In this paper we provide a theoretical background for solving the highlighted problem for a particular class of IF systems. Namely, a theoretical foundation for production of incoming documents' alerts, founded on uncertain classification rules, is discussed. An important functional assumption is that ***alerts are to be stated in quasi-natural languages with linguistic markers for communicating levels of epistemic uncertainty***. In perhaps all languages such linguistic markers exist, usually in a form of well-known and widely used basic modal operators of knowledge *(I know that ...)*, believe *(I believe that ...)* and possibility *(I find it possible/it is possible that ...)*, as well as their possible extensions e.g. *I strongly believe*

Katarzyniak, R., Lorkiewicz, W. and Krejcar, O.
Linguistic Alerts in Information Filtering Systems - Towards Technical Implementations of Cognitive Semantics.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 512-519
ISBN: 978-989-758-187-8

that ... (Nuyts, 2001). The way, in which the natural language operators of epistemic modality should be chosen and used as components of information filtering alerts, is an original contribution of this work, comparing to other works, usually dealing with other classes of language vagueness e.g. (Herrera-Viedma et al., 2004).

The overall organization of the presentation is as follows. In section 2, our original model of knowledge base and basic knowledge management processes, underlying the extraction and application of classification rules to alerts' generation, is presented. In this section a concept of mental language holons is introduced as a key knowledge structure participating to adequate alert's choice and extraction. According to their definition, the holons cover complementary language oriented experience summarizations. In section 3, the so-called theory of modality grounding, originally proposed elsewhere (Katarzyniak, 2005), is applied to define strong and logically consistent support for choosing adequate epistemic modality of alerts. The theory is based on a technical model of so-called cognitive semantics for natural language statements, limited to some scope of quasi-natural language statements, modal in the epistemic sense. The section consists a brief note on the novelty of our approach, considered in respect to technical application of cognitive linguistics. In section 4 a computational example is presented. Finally, section 5 summarizes presented results and points out possible future extensions.

2 MODEL DEFINITION

2.1 Profile of User Information Needs and Filtering Task

The system's user[1] is focused on a given aspect (important to the user) of information stored within the system. This relevant to the user context strictly depends on current user's preferences and is inherently individual. This user's focus represents a set of topics of interest (document's subjects or themes) that are of special importance or significance to the user.

The user's information needs are represented by a set of subjects (also called *themes*) $S = \{sub_1, sub_2, ..., sub_M\}$, being of potential interest to him or her. Moreover, we further assume that information needs represent the sole information that the system captures about the user. Consequently, user's information needs represent the user's profile stored within the system.

We further assume that all of the incoming documents are processed (filtered and indexed) in order to determine whether they cover any of the highlighted topics of interest (user's information needs). The sole purpose of the filtering stage is to identify documents that are significant to the end user. In particular, the goal is to identify documents d that having a complete knowledge about them (for instance through thorough manual examination by an expert) would lead to formulation of basic statements – *"d is about* sub_j*"* or *"d is about* sub_j *and* sub_k*"* (where j is different from k). Recognizing such documents should further lead to generation of adequate alerts by the system, to inform the end user about the appearance of important documents.

Seemingly the introduced form of user profile is extremely trivial, as compared to apparently more complex models of user profiles studied and applied in the field of IF systems e.g. (Brown and Jones, 2001; Shapira et al., 1999; Xu et al., 2011). However, as it turns out in the context of presented approach to generation of linguistic alerts even in such an oversimplified profile posses significant problems. In particular, linguistic approach requires application of unconventional linguistic semantical models that represent solid and adequate theoretical background for technical implementations of cognitive semantics for such linguistic alerts.

Importantly, we should highlight that the aforementioned task of document filtering is highly complex. In automatic approaches it requires a complex and computationally exhaustive procedure that is able to analyse the content of a document and determine it's relevance against the set of identified subjects. Moreover, in some realities a fully automatic approach might not even be available, as such a set of semi-automatic or even manual methods must be utilised. Furthermore, in systems with strict processing time restrictions the filtering process posses significant technical problems, both methodologically and computationally.

2.2 The Repository Databases

The repository consists of two classes of documents: already stored documents $\mathbf{D} = \{d_1, d_2, ..., d_K\}$ with complete descriptions (including a description of their semantic content) stored in a regular database of the repository, and new documents (new arrivals) $\mathbf{D}^{new} = \{d_1^{new}, d_2^{new}, ..., d_{K_{new}}^{new}\}$ with incomplete descriptions of

[1] Due to strict editorial limitations, we focus solely on a single user case. However, in a more practical realisation the proposed approach can be easily extended to a case of multiple users.

their thematic content, thus awaiting off-line semantic analysis.

Formally the repository sub-databases can be described as an information system by Pawlak (Pawlak and Skowron, 2007), tailored to our practical context. Let $Rep = (\mathbf{D} \cup \mathbf{D}^{new}, A, V, \rho)$ be further considered, where $\mathbf{D}$ and $\mathbf{D}^{new}$ are sets of stored documents and new arrivals, respectively, $A = \{w_1, w_2, ..., w_L, sub_1, sub_2, ..., sub_M\}$ is a set of attributes, $V = \bigcup_{a \in A} V_a$, where $V_{w_i} = W_i$ and $V_{sub_i} = \{\varepsilon, 0, 1\}$, is a set of attributes' values, and $\rho : \mathbf{D} \times A \rightarrow V$ is a partial information function.

The partiality of function ρ reflects the extent to which documents are described, regarding their thematic content (their semantics). Namely, it is assumed that $W = \{w_1, w_2, ..., w_L\}$ consists of multivalued attributes, called conditional ones. Values of conditional attributes are usually delivered at document's arrival, as the attributes represent a set of easily computed parameters/characteristics of the document (computed on-line). Contrary, $S = \{sub_1, sub_2, ..., sub_M\}$ consists of attributes, called thematic attributes, representing the content of documents (in respect to a given profile of information needs). Determining the value of thematic attributes requires intensive (both methodological and computational) off-line semantic analysis of the document.

For the sake of clarity and ease of presentation some additional symbols are further introduced.

Namely, for each document $d \in \mathbf{D} \cup \mathbf{D}^{new}$, $\rho_{d|W} : W \rightarrow \bigcup_{i=1}^{L}(W_i)$ is *a conditional-part information function related to document d*, such that for each attribute $x \in W$, $\rho_{d|W}(x) \in W_x$ holds, provided that W_x consists of all possible values of x.

Similarly, for each document $d \in \mathbf{D} \cup \mathbf{D}^{new}$, $\rho_{d|S} : S \rightarrow \{\varepsilon, 0, 1\}$ is *a thematic-part information function related to document d*. However, in this case rules for assigning attribute values differ for $d \in \mathbf{D}$ and $d \in \mathbf{D}^{new}$. Namely for each attribute $x \in S$ and each document $d \in \mathbf{D}$, $\rho_{d|S}(x) = 1$ if and only if document d is indexed as being about subject x. Otherwise $\rho_{d|S}(x) = 0$. At the same time, for each attribute $x \in S$ and $d \in \mathbf{D}^{new}$, the value of x is treated as unknown, what is formally represented by $\rho_{d|S}(x) = \varepsilon$.

2.3 Mental Language Holons as Representation of Subject Distribution

As aforementioned, the introduced IF system is dedicated to analyse incoming documents, regarding individual subjects $sub \in S$ or/and their conjunctions $sub_x \wedge sub_y$, where $sub_x \in S, sub_y \in S, sub_x \neq sub_y$. Results from this analysis may be uncertain predictions, communicated by the means of natural language operators of epistemic modality.

Below an adequate model of database meta-descriptions used in the filtering process is proposed. It's purpose is to enable effective and semantically valid realization of the assumed functional IF system's goal. The model will be fully compatible with an original theory of epistemic modality grounding, partially presented in (Katarzyniak, 2005; Katarzyniak, 2006b; Katarzyniak, 2006a). The main assumption of the theory is that linguistic alerts are inseparably connected to (in a sense *grounded in*) so-called mental language holons . Language holons represent embedded summarization of empirical episodic experiences, i.e., experiences strictly related to particular subjects or their binary conjunctions. In many ways language holons are similar to mental models, known from the cognitive linguistics and psychology (Johnson-Laird, 1985). For the sake of completeness it is worth mentioning that, at the technical level, mental language holons can be treated as complexes of complementary classification rules.

In order to formally capture the latter, the following three retrieval languages are introduced:

$$\begin{aligned} &\mathcal{KS} = \{sub_1, sub_2, ..., sub_M\}, \\ &\mathcal{KB} = \{sub_x \wedge sub_y \mid sub_x, sub_y \in S \wedge x < y\}, \\ &\mathcal{KL} = \{\bigwedge_{i=1}^{L}(w_i = x_i) \mid w_i \in W, x_i \in W_i, i = 1..L\}. \end{aligned} \tag{1}$$

The semantics of retrieval languages is given by following functions:

$$\begin{aligned} &\delta_{|\mathcal{KS}} : \mathcal{KS} \rightarrow 2^{\mathbf{D}}, \\ &\delta_{|\mathcal{KB}} : \mathcal{KB} \rightarrow 2^{\mathbf{D}}, \\ &\delta_{|\mathcal{KL}} : \mathcal{KL} \rightarrow 2^{\mathbf{D} \cup \mathbf{D}^{new}}. \end{aligned} \tag{2}$$

where:

$$\begin{aligned} \delta_{|\mathcal{KS}}(sub) &= \{d \in \mathbf{D} \mid \rho_{d|S}(sub) = 1\}, \\ \delta_{|\mathcal{KB}}(sub_x \wedge sub_y) &= \delta_{|\mathcal{KS}}(sub_x) \cap \delta_{|\mathcal{KS}}(sub_y), \\ \delta_{|\mathcal{KL}}(\bigwedge_{i=1}^{L}(w_i = x_i)) &= \{d \in \mathbf{D} \mid \bigwedge_{i=1}^{L}(\rho_{d|W}(w_i) = x_i)\} \end{aligned} \tag{3}$$

Mental language holons are defined for simple subjects in $\mathcal{KS}$ and conjunctive subjects in $\mathcal{KB}$, in respect to particular conditions from retrieval language $\mathcal{KL}^+ \subseteq \mathcal{KL}$, where the subset (of non-empty conditions) $\mathcal{KL}^+$ is defined as: $\mathcal{KL}^+ = \{k \in \mathcal{KL} \mid \delta_{|\mathcal{KL}}(k) \cap \mathbf{D} \neq \emptyset\}$.

Having defined $\mathcal{KL}^+$, we can introduce two auxiliary symbols class K_i and class extension $EXT(K_i)$. In particular, a class K_i defines a set of indistinguishable (conditional attribute-wise κ_i) already process

documents, whereas class extension $EXT(K_i)$ defines a set of indistinguishable (conditional attribute-wise κ_i) all documents. Namely, if (and only if) $|\mathcal{KL}^+| = Q \geq 1$ and $\mathcal{KL}^+ = \{\kappa_1, \kappa_2, ..., \kappa_Q\}$, then for $i = 1..Q$,

$$\begin{aligned} K_i &= \delta_{|\mathcal{KL}}(\kappa_i) \cap \mathbf{D}, \\ EXT(K_i) &= \delta_{|\mathcal{KL}}(\kappa_i) \cap \mathbf{D}^{new}. \end{aligned} \tag{4}$$

For each $sub \in S$ and $\kappa_i \in \mathcal{KL}^+$, the (simple subject) mental language holon is given as a vector *simholon*:

$$simholon[\kappa_i, sub, \lambda_A^+(sub), \lambda_A^-(sub)], \tag{5}$$

where

$$\begin{aligned} \lambda_A^+(sub) &= \frac{|\delta_{|\mathcal{KS}}(sub) \cap K_i|}{|K_i|}, \\ \lambda_A^-(sub) &= \frac{|(\mathbf{D} \setminus \delta_{|\mathcal{KS}}(sub)) \cap K_i|}{|K_i|}. \end{aligned} \tag{6}$$

For each conjunctive subject $(sub_x \wedge sub_y) = sub_{xy} \in \mathcal{KB}$ and $\kappa_i \in \mathcal{KL}^+$, the (conjunctive subject) mental language holon is given as a vector *conholon*:

$$\begin{aligned} conholon[\kappa_i, sub_{xy}, \lambda_C^{++}(sub_{xy}), \lambda_C^{+-}(sub_{xy}), \\ \lambda_C^{-+}(sub_{xy}), \lambda_C^{--}(sub_{xy})], \end{aligned} \tag{7}$$

where

$$\begin{aligned} \lambda_C^{++}(sub_{xy}) &= \frac{|\delta_{|D}(sub_x) \cap \delta_{|D}(sub_y) \cap K_i|}{|K_i|}, \\ \lambda_C^{+-}(sub_{xy}) &= \frac{|\delta_{|D}(sub_x) \cap (\mathbf{D} \setminus \delta_{|D}(sub_y)) \cap K_i|}{|K_i|}, \\ \lambda_C^{-+}(sub_{xy}) &= \frac{|(\mathbf{D} \setminus \delta_{|D}(sub_x)) \cap \delta_{|D}(sub_y) \cap K_i|}{|K_i|}, \\ \lambda_C^{--}(sub_{xy}) &= \frac{|(\mathbf{D} \setminus \delta_{|D}(sub_x)) \cap (\mathbf{D} \setminus \delta_{|D}(sub_y)) \cap K_i|}{|K_i|}. \end{aligned} \tag{8}$$

From the pragmatic point of view, mental language holons are higher level summarizations (semantic generalizations) of relative share of complementary bodies of experiences, related to particular subjects (or their conjunctions). The whole repository of language holons, available to IF system's processes and, in particularly to alerts production procedures, is given as follows:

$$\begin{aligned} HOLONS &= SIMHOLONS \cup CONHOLONS, \\ SIMHOLONS &= \{simholon[\kappa, x, \lambda_A^+(x), \lambda_A^-(x)] \\ &\quad | \ \kappa \in \mathcal{KL}^+, x \in \mathcal{KS}\}, \\ CONHOLONS &= \{conholon[\kappa, x, \lambda_C^{++}(x), \lambda_C^{+-}(x), \lambda_C^{-+}(x), \\ &\quad \lambda_C^{--}(x)] \mid \kappa \in \mathcal{KL}^+, x \in \mathcal{KB}\}. \end{aligned} \tag{9}$$

3 ALERTS PRODUCTION

3.1 Alerts and their Semantic Proto-forms

Examples of possible structure and content of alerts, considered in our research, are given as follows:

IF SYSTEM ALERT: There is a new [document: x]. I believe it is about [subject: sub]. You may be interested in reading it!

IF SYSTEM ALERT: Documents [documents: $x_1, ..., x_k$] are new. It is possible that they are about [subjects: sub_x and sub_y]. Should I put them on your pending list?

IF SYSTEM ALERT: There is a new [document: x] worth of being looked at. I believe it is about [subject: sub_x], but not about [subject: sub_y]. According to what I know about your interests, the first issue may be of interest to you. Should I put the document to your working box? Please, answer [YES/NO]!

IF SYSTEM ALERT: Among others, the following documents: $x_1, ..., x_k$ have been received from [source: source], too. I believe they are not about [subject:sub] which you pointed at as your main issue. Whether, despite this shall I put them on your pending list? Please, answer [YES/NO]!

IF SYSTEM ALERT: It is possible that the following [incomings: $x_1, ..., x_L$] deal with [subject:sub], which is on your list of interests. Are you interested in reading them before turning them to our central document base? Please, answer [YES/NO]!

The structure of alerts fully depends on designer's choice and, obviously, it should reflect favoured modes and preferences of particular user (users' group) interactions. In our case the alerts are represented (communicated) in a natural language, which is a partially controlled version of actual language. In advanced multimedia systems the alerts can be vocalized, too.

The common feature of the above examples is their underlying sense. Namely, regardless of their form (individual document vs. group of documents, simple subject vs. conjunctive subject), they all are founded on the same propositional aspect: being about or not being about a particular simple subject (or conjunction of simple subjects). Moreover, For $x \in D \cup D^{new}$ and $sub \in \mathcal{KS} \cup \mathcal{KB}$, each example is

originally created as instantiation (concretization) of one of the following basic linguistic proto-forms:
knowing([document(s):x] is about [subject(s):sub])
believing([document(s):x] is about [subject(s):sub])
possible([document(s):x] is about [subject(s):sub])
or another proto-form, complementary to the above enumerated ones.
It is worth of mentioning that for a fixed document *x* and a fixed subject *sub* (a simple subject or a binary conjunction of simple subjects) one and only one proto-form should be instantiated as proper representation of epistemic state. Namely, such constraint follows from common sense, natural language pragmatics rule, saying that knowing, believing and finding something only as possible (in the epistemic sense) are mutually exclusive, different states of the same mental epistemic attitude. Thus, in our research an adequate extraction of natural language alerts from IF system's knowledge base (or more strictly: proper and adequate choice and further instantiation of proto-form) becomes a fundamental issue to be elaborated, on both technical and theoretical levels.

In conclusion, similarly to other natural language statements, three aspects of alerts need to be taken into account: *propositional element, modality, and temporal frame*. As it has just been mentioned above, the propositional element is given by predication, which on written (or vocalized) level is referred to by elements of sets $\mathcal{KS}$ and $\mathcal{KB}$. The alerts' temporal dimension is quite apparent. Namely, they are stated in the present grammatical time. A more problematic issue is the alerts' modality choice, which in our case should reflect a kind of epistemic uncertainty of IF system, itself. An important question, of both theoretical and technical nature, is how to properly choose adequate modality markers, in order to extend written (or vocalized) representation of predication (applied to incoming documents). This question is strongly supported by an original theory of grounding of modal epistemic statements, briefly presented below.

3.2 Applying the Theory of Epistemic Modality Grounding to Alerts' Production

The decision rules for proper choice of an adequate modal proto-form, its instantiation (and further presentation to an end user in a written and/or vocalized form) follow from an original theory of grounding, presented elsewhere. Namely, for the case of simple subject-based predication the introductory theoretical results can be found in (Katarzyniak, 2005), for binary conjunctive subject-based predication in (Katarzyniak, 2006b; Katarzyniak, 2006a).

It is assumed in the theory (following multiple models of language production (Evans and Green, 2006; Stachowiak, 2013; Wlodarczyk, 2013)) that particular epistemic operators of modality are related to summarized empirical experience, supporting related language proto-forms. However, these proto-forms are never stored and processed as separate entities, for they are conceptually (mentally) related to their complementary counterparts. In particular, such complexes of complementary proto-forms constitute linguistic holons, which in our technical approach are strongly related to the concept of mental language holons, defined in the previous sections. In consequence, to each linguistic proto-form, always related to one and only one part of a relevant mental language holon, certain intensity of summarized (embodied) experience of a subject (or binary conjunctive subject) is assigned. In the theory of grounding this intensity is numerically represented by the relative grounding strength.

According to the theory of simple modalities grounding, the proper choice of adequate linguistic proto-form is possible if and only if a proper system of the so-called modality thresholds is applied (and technically realized in a system). In our case the system needs to consist of two interrelated sub-systems of thresholds $\{\lambda^{KS}_{Know}, \lambda^{KS}_{maxBel}, \lambda^{KS}_{minBel}, \lambda^{KS}_{maxPos}, \lambda^{KS}_{minPos}\}$ and $\{\lambda^{\wedge}_{Know}, \lambda^{\wedge}_{maxBel}, \lambda^{\wedge}_{minBel}, \lambda^{\wedge}_{maxPos}, \lambda^{\wedge}_{minPos}\}$, for effective control of simple-subject predication instantiation and conjunctive subject predication instantiation, respectively.

An interesting result from the theory of grounding, for the practice perhaps the most important one, is that the system of modality thresholds cannot be freely chosen. Namely, in order to guarantee common sense consistency of (written and verbal) language behaviour the system of modality thresholds has to fulfil some predefined set of requirements, accepted in the theory of grounding, as a reflection of common sense pragmatics applied in actual contexts to natural language operators of knowledge, belief, and possibility. The fact that written and/or verbal behaviour, produced by a technical system based on the theory of grounding, is actually consistent, from the semiotic and pragmatic point of view, can be analytically proved and verified[2].

Moreover, within the numerical scope which is permissible according to the theory of grounding, values for thresholds can be chosen in an arbitrary manner (Katarzyniak, 2005). However, for the case of

[2]Some of the results can be found in (Katarzyniak, 2005; Katarzyniak, 2006b; Katarzyniak, 2006a).

populations of artificial agents it is be possible to obtain them from computationally realized processes of artificial language semiosis (Lorkiewicz et al., 2011).

In order to omit deeper discussion of the theory of grounding (outside of the scope of this work) we further present an original application of the theory to basic rules definition for modal alerts' acceptability and adequacy. The fundamental assumption is that a given modal alert can be produced (by IF system) if and only if its underlying linguistic proto-form is well-grounded in IF system's knowledge base. It means, too, that in this practical context, for a certain alert *being well grounded* is equivalent to *adequately describing a related IF system's state of knowledge about possibility of a certain document* $d \in D \cup D^{new}$ *to deal with a certain subject* $sub \in \mathcal{KS} \cup \mathcal{KB}$. In particular, for any document $d \in D^*$, $d \in EXT(K_i)$, and $sub \in \mathcal{KS}$, the following set of so-called grounding relations constitute the theoretical foundation of IF alerting processes:

$$simholon[\kappa_i, sub, \lambda_A^+(sub), \lambda_A^-(sub)]$$
$$\models_G \textbf{possible}([d] \text{ is about } [sub])$$

holds if and only if $\lambda^{KS}_{\textbf{minPos}} \leq \lambda_A^+(sub) < \lambda^{KS}_{\textbf{maxPos}}$.

$$simholon[\kappa_i, sub, \lambda_A^+(sub), \lambda_A^-(sub)]$$
$$\models_G \textbf{believing}([d] \text{ is about } [sub])$$

holds if and only if $\lambda^{KS}_{\textbf{minBel}} \leq \lambda_A^+(sub) < \lambda^{KS}_{\textbf{maxBel}}$.

$$simholon[\kappa_i, sub, \lambda_A^+(sub), \lambda_A^-(sub)]$$
$$\models_G \textbf{knowing}([d] \text{ is about } [sub])$$

holds if and only if $\lambda_A^+(sub) = \lambda^{KS}_{\textbf{Know}} = \mathbf{1}$.

Rather obviously, complementary alerts on *document* $d \in D^*$ *not being about a particular subject* $sub \in \mathcal{KS}$, are produced with respect to the next three definitions:

$$simholon[\kappa_i, sub, \lambda_A^+(sub), \lambda_A^-(sub)]$$
$$\models_G \textbf{possible}([d] \text{ is } \textbf{not} \text{ about } [sub])$$

holds if and only if $\lambda^{KS}_{\textbf{minPos}} \leq \lambda_A^-(sub) < \lambda^{KS}_{\textbf{maxPos}}$.

$$simholon[\kappa_i, sub, \lambda_A^+(sub), \lambda_A^-(sub)]$$
$$\models_G \textbf{believing}([d] \text{ is } \textbf{not} \text{ about } [sub])$$

holds if and only if $\lambda^{KS}_{\textbf{minBel}} \leq \lambda_A^-(sub) < \lambda^{KS}_{\textbf{maxBel}}$.

$$simholon[\kappa_i, sub, \lambda_A^+(sub), \lambda_A^-(sub)]$$
$$\models_G \textbf{knowing}([d] \text{ is } \textbf{not} \text{ about } [sub])$$

holds if and only if $\lambda_A^-(sub) = \lambda^{KS}_{\textbf{Know}} = \mathbf{1}$.

Obviously, similar set of definitions, for $d \in D^*$, $d \in EXT(K_i)$, and $(sub_x \wedge sub_y) = sub_{xy} \in \mathcal{KB}$, can also be formulated and used, if needed. However, in such case another mental language holons must be referred to:

$$conholon[\kappa_i, sub_{xy}, \lambda_C^{++}(sub_{xy}), \lambda_C^{+-}(sub_{xy}), \lambda_C^{-+}(sub_{xy}), \lambda_C^{--}(sub_{xy})]$$
$$\models_G \textbf{possible}([d] \text{ is about } [sub_x] \text{ and } [sub_y])$$

holds if and only if $\lambda^{\wedge}_{\textbf{minPos}} \leq \lambda_C^{++}(sub) < \lambda^{\wedge}_{\textbf{maxPos}}$.

$$conholon[\kappa_i, sub_{xy}, \lambda_C^{++}(sub_{xy}), \lambda_C^{+-}(sub_{xy}), \lambda_C^{-+}(sub_{xy}), \lambda_C^{--}(sub_{xy})]$$
$$\models_G \textbf{believing}([d] \text{ is about } [sub_x] \text{ and } [sub_y])$$

holds if and only if $\lambda^{\wedge}_{\textbf{minBel}} \leq \lambda_C^{++}(sub) < \lambda^{\wedge}_{\textbf{maxBel}}$.

$$conholon[\kappa_i, sub_{xy}, \lambda_C^{++}(sub_{xy}), \lambda_C^{+-}(sub_{xy}), \lambda_C^{-+}(sub_{xy}), \lambda_C^{--}(sub_{xy})]$$
$$\models_G \textbf{knowing}([d] \text{ is about } [sub_x] \text{ and } [sub_y])$$

holds if and only if $\lambda_C^{++}(sub_{xy}) = \lambda^{\wedge}_{\textbf{Know}} = \mathbf{1}$.

For purely editorial reasons, we do not deal with the complementary conjunctive alerts, i.e., alerts on new documents *being about* [sub_x *and not* sub_y], [*not* sub_x *and* sub_y], [*not* sub_x *and not* sub_y]. It is quite obvious that they have to be verified in a similar way, but against values of $\lambda_C^{+-}(sub_{xy})$, $\lambda_C^{-+}(sub_{xy})$, and $\lambda_C^{--}(sub_{xy})$, respectively.

3.3 A Brief Note on Cognitive Semantics

The novelty of our approach to the generation of quasi-natural language alerts falls outside of previous linguistic models. Namely, it is an original proposal consistent with cognitive linguistics (Evans and Green, 2006) and interactive linguistics (Wlodarczyk, 2013) paradigms. Both of them refer our work to the concept of cognitive semantics (Talmy, 2000), which describes the way a particular natural language sentence embraces the pre-linguistic knowledge corpora accessible to minds of a communicative agent. Obviously, in our *R&D* context the communicating subjects are IF systems.

Cognitive semantics is always characterised by high specificity, because in each case it reflects pragmatics and meaning of a very narrow class of linguistic phenomena. In our model this specificity is apparently visible in internally related and complex structure of mental language holons. A proposal of how to realize the cognitive semantics of alerts in our IF system should be treated as the most original contribution of the model.

4 COMPUTATIONAL EXAMPLE

In this section we introduce a basic example that illustrates the entire process of generating linguistic alerts in IF systems. For the sake of simplicity let us assume an elementary information systems comprised

of a document repository consisted of 10 processed documents $D = \{d_1, d_2, ..., d_{10}\}$ and 3 new documents $D^{new} = \{d_{11}, d_{12}, d_{13}\}$ that are evaluated based on a set of 4 conditional attributes $W = \{w_1, w_2, w_3, w_4\}$. Further, let us assume that user's information needs are limited to two subjects $S = \{sub_1, sub_2\}$. Consequently, the set of all attributes available in the system is defined as $A = \{w_1, w_2, w_3, w_4, sub_1, sub_2\}$. Furthermore, let the domains of the introduced attributes be given as follows, $W_1 = W_2 = W_3 = W_4 = \{A, B, C\}$.

Documents stored in the document repository are processed. In particular, each document is analysed by a set of indexing mechanisms (or other processing mechanisms) that are able, based on the document content and structure, to assign values for each of the conditional attributes. Further, information about each document's subject is determined and stored. As such the information function of the repository is determined, i.e., attribute–value mapping, as given in Table.1.

Focusing on three simple classes κ_1, κ_2, and κ_3, given as $\kappa_1 = \{(w_1, B), (w_2, A), (w_3, A), (w_4, A)\}$, $\kappa_2 = \{(w_1, C), (w_2, C), (w_3, A), (w_4, B)\}$, and $\kappa_3 = \{(w_1, B), (w_2, B), (w_3, C), (w_4, A)\}$, we can determine three non-empty clusters of documents $K_1 = \{d_1, d_2, d_3, d_6\}$, $K_2 = \{d_4, d_5, d_7, d_{10}\}$, $K_3 = \{d_8, d_9\}$ and their extensions $EXT(K_1) = \{d_{11}\}$, $EXT(K_2) = \{d_{12}\}$, $EXT(K_3) = \emptyset$. It must to be mentioned that one of the newly received documents, namely d_{13}, does not belong to any of these sets. This fact will be commented in the final remarks section.

Resulting summarization of data is represented by the following set of holons HOLONS = SIMHOLONS ∪ CONHOLONS:

$$\begin{aligned} SIMHOLONS = \{ & \\ & simholon[\kappa_1, sub_1, 0.25, 0.75], \\ & simholon[\kappa_1, sub_2, 1.00, 0.00], \\ & simholon[\kappa_2, sub_1, 1.00, 0.00], \\ & simholon[\kappa_2, sub_2, 0.25, 0.75], \\ & simholon[\kappa_3, sub_1, 0.50, 0.50], \\ & simholon[\kappa_3, sub_2, 0.00, 1.00]\}. \end{aligned} \tag{10}$$

$$\begin{aligned} CONHOLONS = \{ & \\ & conholon[\kappa_1, sub_1 \wedge sub_2, 0.25, 0.50, 0.25, 0.00], \\ & conholon[\kappa_2, sub_1 \wedge sub_2, 0.25, 0.75, 0.00, 0.00], \\ & conholon[\kappa_3, sub_1 \wedge sub_2, 0.00, 0.50, 0.00, 0.50]\}. \end{aligned} \tag{11}$$

Having the relative grounding strength computed and stored in each holon, we can now determine all proto-forms, for the new arrivals from non-empty extensions $EXT(K_1)$ and $EXT(K_2)$.

To give an example, simple subjects will be considered. Let modality thresholds be set up to following values $\lambda^{KS}_{Know} = \lambda^{KS}_{maxBel} = 1$, $\lambda^{KS}_{minBel} = \lambda^{KS}_{maxPos} = 0.60$, and $\lambda^{KS}_{minPos} = 0.20$. These values are not accidental. Namely, they have been chosen taking into account theorems from the theory of grounding simple modalities (Katarzyniak, 2005). It follows that the threshold values should preserve consistency of sets of grounded proto-forms with common sense interpretation. Below we provide examples of well-grounded grounded proto-forms:

Table 1: Processed repository of documents.

	w_1	w_2	w_3	w_4	s_1	s_2
d_1	B	A	A	A	1	1
d_2	B	A	A	A	0	1
d_3	B	A	A	A	0	1
d_4	C	C	A	B	1	0
d_5	C	C	A	B	1	0
d_6	B	A	A	A	0	1
d_7	C	C	A	B	1	0
d_8	B	B	C	A	0	0
d_9	B	B	C	A	1	0
d_{10}	C	C	A	B	1	1
d_{11}	B	A	A	A	ε	ε
d_{12}	C	C	A	B	ε	ε
d_{13}	B	B	C	B	ε	ε

- **possible**($[d_{11}]$ is about $[sub_1]$) AND **possible**($[d_{11}]$ is **not** about $[sub_1]$)
- **believing**($[d_{11}]$ is about $[sub_2]$), BUT STILL **possible**($[d_{11}]$ is **not** about $[sub_2]$)
- **knowing**($[d_{12}]$ is about $[sub_2]$)

It is worth of mentioning that these proto-forms are logically consistent, which is ensured by the proper choice of modality thresholds. A possible natural language alert founded on the established proto-forms is:
IF SYSTEM ALERT: There is a new [document: doc_{12}] available. I believe it is about [subject: sub_2]. You may be interested in reading it!.

5 FINAL REMARKS

The theoretical foundation for designing and implementing interactive IF systems is proposed in this paper. The desirable common sense consistency of quasi-natural language alerts is ensured by the application of a theory of epistemic modality grounding, introduced elsewhere. The proposal substantially differs from previous models of similar alerts generation.

The proposed model of linguistic alerts choice and production is supported by a simple computational methodology and a naive model of uncertain classification rules. Alternative and more sophisticated approaches are possible (and required) e.g. for the way sets K_i and $EXT(K_i)$ are determined. Obviously,

complete final implementation need to cover the missing case of document d_{13}, either.
The introduced model supports effective design and implementation of modern interactive and mobile tools for alerting end users about newly received objects of potential interests, in both written and vocalized modal natural languages.

ACKNOWLEDGEMENTS

This work was realized under research cooperation between Wrocław University of Technology (Faculty of Computer Science and Management - Internal Grant No. S50198 K0803) and Hradec Králové University (Center for Basic and Applied Research, Faculty of Informatics and Management SP-FIM-2016 - Smart Solutions for Ubiquitous Computing Environments).

REFERENCES

Brown, P. J. and Jones, G. J. F. (2001). Context-aware retrieval: Exploring a new environment for information retrieval and information filtering. *Personal and Ubiquitous Computing*, 5(4):253–263.

Evans, V. and Green, M. (2006). *Cognitive linguistics: An introduction*. Edinburgh University Press.

Hanani, U., Shapira, B., and Shoval, P. (2001). Information filtering: Overview of issues, research and systems. *User Modelling and User-Adapted Interaction*, 11(3):203–259.

Herrera-Viedma, E., Herrera, F., Martínez, L., Herrera, J. C., and López, A. G. (2004). Incorporating filtering techniques in a fuzzy linguistic multi-agent model for information gathering on the web. *Fuzzy Sets and Systems*, 148(1):61–83.

Johnson-Laird, P. N. (1985). *Towards a Cognitive Science of Language, Inference, and Consciousness*. Cambridge University Press, Cambridge.

Katarzyniak, R. (2005). On some properties of grounding simple modalities. *Systems Science*, 31(3):59–86.

Katarzyniak, R. (2006a). On some properties of grounding nonuniform sets of modal conjunctions. *Int. Journal of Applied Mathematics and Computer Science*, 16(3):399–412.

Katarzyniak, R. (2006b). On some properties of grounding uniform sets of modal conjunctions. *Journal of Intelligent and Fuzzy Systems*, 17(3):209–218.

Lorkiewicz, W., Popek, G., Katarzyniak, R., and Kowalczyk, R. (2011). Aligning Simple Modalities in Multiagent System. In *Proc. ICCCI 2011*, volume 6923, pages 70–79. LNAI.

Nuyts, J. (2001). *Epistemic Modality, Language, and Conceptualization: A Cognitive-pragmatic Perspective*.

Pawlak, Z. and Skowron, A. (2007). Rudiments of rough sets. *Information Sciences*, 177(1):3–27.

Shapira, B., Shoval, P., and Hanani, U. (1999). Experimentation with an information filtering system that combines cognitive and sociological filtering integrated with user stereotypes. *Decision Support Systems*, 27:5–24.

Stachowiak, F. J. (2013). Tracing the role of memory and attention for the meta-informative validation of utterances. In Wodarczyk, A. and Wodarczyk, H., editors, *Meta-informative Centering in Utterances: Between Semantics and Pragmatics*, pages 121–142. John Benjamins Publishing Co., Amsterdam.

Talmy, L. (2000). *Toward a Cognitive Semantics*. MIT Press, Cambridge, MA.

Wlodarczyk, A. (2013). Grounding of the meta-informative status of utterances. In Wlodarczyk, A. W. and H., editors, *Meta-informative Centering in Utterances: Between Semantics and Pragmatics*, pages 41–58. John Benjamins Publishing Co., Amsterdam.

Xu, M., Ong, V., Duan, Y., and Mathews, B. (2011). Intelligent agent systems for executive information scanning, filtering and interpretation: Perceptions and challenges. *Information Processing and Management*, 47(2):186–201.

A Big Data based Smart Evaluation System using Public Opinion Aggregation

Robin G. Qiu[1], Helio Ha[1], Ramya Ravi[1], Lawrence Qiu[2] and Youakim Badr[3]
[1]*Big Data Lab, Penn State University, Engineering Division, Malvern, U.S.A.*
[2]*School of EE & CS, Penn State University, University Park, Pennsylvania, U.S.A.*
[3]*LIRIS-CNRS, Département Informatique, INSA de Lyon, Lyon, France*
{robinqiu, hvh5248, rxr303, lyq5024}@psu.edu, youakim.badr@insa-lyon.fr

Keywords: Big Data, Smart Evaluation System, Higher Education Services, Rankings, Ranking System, Sentiment Analysis, Public Opinions.

Abstract: Assessing service quality proves very subjective, varying with objectives, methods, tools, and areas of assessment in the service sector. Customers' perception of services usually plays an essential role in assessing the quality of services. Mining customers' opinions in real time becomes a promising approach to the process of capturing and deciphering customers' perception of their service experiences. Using the US higher education services as an example, this paper discusses a big data-mediated approach and system that facilitates capturing, understanding, and evaluation of their customers' perception of provided services in real time. We review such a big data based framework (Qiu et al., 2015) in support of data retrieving, aggregations, transformations, and visualizations by focusing on public ratings and comments from different data sources. An implementation with smart evaluation services is mainly presented.

1 INTRODUCTION

Service quality is well recognized as the overall perception of the services that results from comparing the service provider's performance with the general expectations of customers of how the service provider in that industry should perform. Challengingly, assessing service quality proves very subjective, varying with objectives, methods, tools, and areas of consideration in the service sector. Regardless of how service providers think about their provided services, frequently to a customer, it is the encounter of a service or 'moment of truth' that defines the service. In other words, it is the experience that customers perceived from their encountered services subjectively defines service quality (Qiu, 2014). Therefore, customers' perceptions of experienced services play an essential role in assessing the quality of consumed services. Correspondingly, mining customer or public opinions in real time becomes a promising approach to the process of capturing customers' perceptions of their service experiences (Meyer and Schwager, 2007; Labrecque et al., 2013).

Education has been one of main services in the US service sector for many decades. Ensuring that the US education service performs well is one of top nation's priorities. The higher education particularly draws much attention from a variety of stakeholders, from students, parents, employers, the government, to college administrators and boards of directors. Hence, finding a reliable method of knowing how the US higher education as a whole or an individual college is performing is necessary. Over several decades, there have been a variety of ranking systems that in different perspectives provide assessments of education services on higher education nationally or internationally (Harvey, 2008; Bergseth et al., 2014). A few well known ranking systems include the US News & World Report (USNWR), the Times Higher Education (THE) from the United Kingdom, and the Academic Ranking of World Universities (ARWU) from China's Shanghai Jiao Tong University (SJTU) (Huang and Qiu, 2016).

Regardless of ranking system or metrics, it is typical to utilize service quality factors that are subjectively selected and weighted. As a result, the provided rankings' objectivity and impartiality become worrisome and sometimes confusing and misleading (MIT, 2011). To some extent, a quantitative and model-driven method to

Qiu, R., Ha, H., Ravi, R., Qiu, L. and Badr, Y.
A Big Data based Smart Evaluation System using Public Opinion Aggregation.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 520-527
ISBN: 978-989-758-187-8

computationally generate ranking factors' weightings in a ranking system can help address the objectivity and impartiality issues in its enabled rankings (Huang and Qiu, 2016). The method bears an acronym of HESSEM, i.e., Higher Education System oriented Structural Equation Modeling. Because selecting ranking factors for a ranking system is also subjective, thus it is desirable for a ranking system to allow ranking factors to be easily adjusted, i.e., removed from or added into the ranking system. Promisingly, HESSEM allows ranking factors to be easily changed whenever needed. However, it is never easy to identify new factors impacting on rankings and then capture sufficient data for the identified factors (Qiu et al., 2015).

Gathering customers' perceptions of their experienced services is still the most pervasive and dominative means for service organizations to decipher the quality of services provided by the organizations although there have been a lot of changes in terms of tools and instruments used in capturing and understanding customers' perspectives over the years. Periodically conducting customer surveys and interviews has been popularly adopted in the fields of marketing and after-sale services, aimed at enhancing product designs, prioritizing engineering and marketing efforts, and improving after-sale services. Today, with the help of digital media, mobile and pervasive computing, and significantly enhanced tools and methods to capture and understand customer interaction and behavior in its deepen and refined granularity, traditional approaches have been evolved and substantially augmented by incorporating social data along with traditional data sources to provide a complete picture of customers (Ahlquist and Saagar, 2013; Labrecque et al., 2013; Qiu, 2014; Qiu et al., 2014). As a result, capturing and understanding not only cross-sectional and longitudinal but also real time and comprehensive customers' perceptions of services become practically implementable.

"People-centric sensing will help drive this trend by enabling a different way to sense, learn, visualize, and share information about ourselves, friends, communities, the way we live, and the world we live in." (Campbell et al., 2008) Thus, it is worthy to explore a way to collect and aggregate public opinions to enhance assessment of service quality (Qiu et al., 2015). This paper uses the US higher education as a service example to show how big data-mediated public opinion aggregation can be well applied to augmenting the assessment of service quality. Please keep in mind, a system of computing rather than a service quality modeling approach is presented in the remaining paper.

The remaining paper is organized as follows. Section 2 briefly reviews a framework for capturing and visualizing public options that has been used to develop a ranking module, part of the Leveraging Innovative Online Networks to Learn Education Networks and Systems (LIONLENS) research project by the authors. Section 3 then discusses how the LIONLENS enables the core and fundamental computing supports necessary for the realization of aggregating and visualizing public opinions and sentiment trends on the US higher education in its ranking module. Finally, a brief conclusion for this paper is given in Section 4.

2 A FRAMEWORK FOR CAPTURING AND VISUALIZING PUBLIC OPINIONS

To the service provider of a service system, capturing, understanding, and controlling the interaction among all the stakeholders of a service system plays an essential role in designing, developing, and managing the service system (Qiu, 2014). A ranking system undoubtedly is a service system. Thus, real time capturing and understanding public perceptions or opinions become necessary for the development of a desirable and reliable ranking system, which would not only meet the needs of the public but also be well aligned with the long-term goals of ranking service providers. Bearing this understanding in mind, a framework for developing the LIONLENS including capturing and visualizing public opinions has been proposed (Qiu et al., 2015), which is graphically illustrated in Figure 1. Technically and financially, the proposed framework allows the LIONLENS to be modularly and then gradually developed while evolving over time (Qiu, 2014).

As shown in Figure 1, the emerging big data technologies, widely adopted mobile computing, and social media can be fully applied and leveraged to facilitate the process of monitoring and deciphering the public's acceptance and colleges' performance in real time (Qiu et al., 2015). The highlights of two different perspectives of the proposed framework in Figure 1 can be briefed explained as follows:

- From the systems perspective: an education system consists of people, technologies, resources, and education service products that

can generate respective values for all stakeholders through service provision. To evaluate education service quality, different aspects of data on the system and the public perception of its provided education must be captured and deciphered. Indeed, in addition to using traditional data source approaches, distributed and mobile computing systems and applications have been leveraged so that data and information on college education services, students enrollment profiles, faculty performances, school facilities, campus life, etc. can be effectively captured, retrieved, and archived, college by college and/or colleges as a whole. Technically, the implementations of existing ranking systems differs significantly from each other. However, there is no significant difference in terms of data collection tools and methods adopted by existing ranking systems.

- From the analytical perspective: valid and effective modeling methodologies should be applied to not only enable ranking services, but also uncover the insights from the collected data and information and ultimately provide prompt guidance for administrators to take action for positive changes. Ranking systems vary with adopted modeling methodologies, computing technologies and implementations, and operational models. As a result, ranking and administrative services enabled by the ranking systems could be descriptive, predictive, and/or prescriptive.

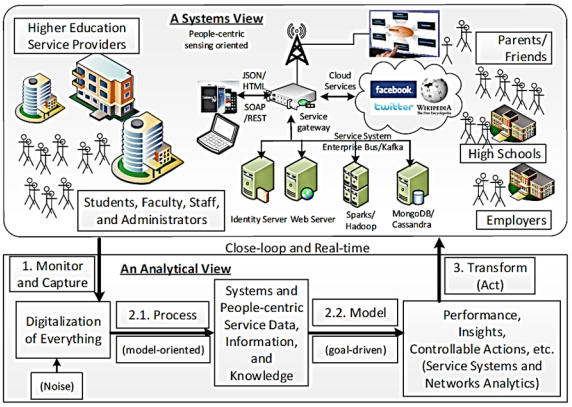


Figure 1: A framework for developing the LIONLENS.

With the advent of the Internet and mobile computing, voluminous and various data on higher education can be retrieved and mined from the Internet and social media. In other words, as such data becomes richer and richer, the list of ranking (or service quality) indicators should become easier to be adjusted (i.e., added or removed) whenever necessary. For instance, the inputs from the public are vital for ranking systems. Therefore, public ratings and comments must be taken into consideration so that a ranking system can evolve to better meet the needs of stakeholders. Figure 2 shows the logic flows of the design and implementation of the ranking module in the LIONLENS, which gets enhanced by incorporating public ratings and comments.

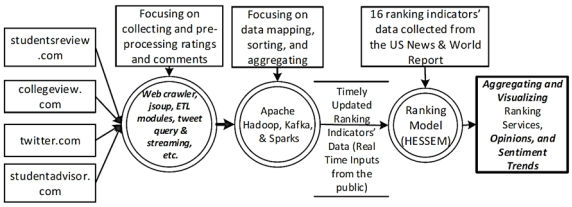


Figure 2: Monitoring, capturing, and visualizing colleges' performance and public opinions.

As discussed earlier, this paper focuses on the discussion of a systems and computing approach to enhancing service quality assessment for the US higher education. In other words, using the systems perspective we aim to show how big data-mediated public opinion aggregation can be practically applied to addressing service assessment problems. In particular, we show how big data technologies, mobile computing, and social media can be fully leveraged to facilitate the process of monitoring, capturing, and visualizing colleges' performance and public opinions on education service quality in real time. As shown in Figure 2, the process of adjusting ranking factors is enhanced by including public opinions' sentiment analysis. In practice, public opinions can be captured, retrieved, and analyzed from websites and online media including twitters.

We have developed HESSEM - a quantitative and model-driven ranking model to evaluate higher education service quality in the US. Using collected data, we applied structural equation modeling to systematically determine ranking factor weights for assessing education service quality and performance of the US higher education (Huang and Qiu, 2016). By extending the brief discussion presented in our previous paper (Qiu et al., 2015), this paper in great detail discusses how the public textual inputs can be captured and filtered, and aggregated to enhance educational service quality assessment. Therefore, in the next section we explain technically and functionally the core computing components and algorithms applied in this study.

3 BIG DATA-MEDIATED FUNCTIONAL SUPPORTS

As mentioned earlier, in this paper we focus on presenting technically and functionally the data flows and computing components deployed in the LIONLENS that support the retrieving and aggregating of the public opinions (Figure 3).

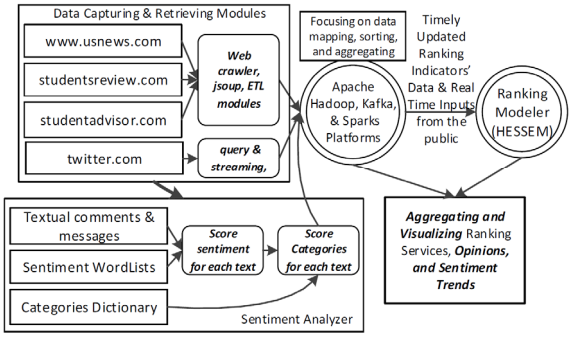


Figure 3: Functional flows and components in support of retrieving and aggregating the public opinions.

As highlighted in Figure 3, five main computing components in support of the ranking services enabled by the LIONLENS, which are briefly introduced as follows:

- Data capturing & retrieving modules: web crawler, data extract-transform-load, and tweets query and streaming modules are applied and developed for retrieving and pre-processing data from different data sources over the Internet.
- Sentiment analyzer: Collected data and information are saved as files that are transformed and analyzed to generate sentiment scores, indicating service performance trends over time.
- Big data computing clusters or platforms: Apache Hadoop & Sparks platform technologies based on Lambda architecture is used to aggregate, consolidate, and archive the captured and pre-processed data.
- Ranking modeler: HESSEM is adopted for generating rankings on a daily basis based on archived and on-going, newly collected and updated data.
- Visualization modules: interactive web interfaces are developed and deployed to allow end users to visualize aggregated public opinions and sentiment trends on the higher education in the US.

3.1 Data Capturing & Retrieving Modules

To demonstrate how data and information on education services can be used to enhance service quality modeling, we developed web crawler, data extract-transform-load, and tweets query and streaming modules to crawl across a list of selected websites and retrieve public comments and tweets. As numerous information retrieval tools and libraries are available over the Internet, these data retrieving modules can be easily customized for other websites. We use the studentadvisor.com as an example to show how public ratings and comments are captured, pre-processed, and utilized in this project.

The studentadvisor.com website allows the public to post their comments and ratings on any colleges in the US. Ratings spanning over 6 categories from overall, academics, campus facility, sports, student life, surrounding area, to worth the money are Likert-scale based, from 1 to 5. Public comments namely 'The Good', 'The Bad', and 'WouldIdoItAgain' are then text based. As soon as web pages are downloaded, the targeted data including ratings and comments are extracted and transformed. To ensure that retrieved data will be readily accepted by the big data platforms, the transformed data for each college is loaded into a corresponding CSV file as illustrated in Figure 4.

URL	PostTime	Overall	Academics	Campus.Fa	Sports	Student.Li	Surroundi	Worth.the	Comment1	Comment2
stanfordu	August 1'	5	5	5	5	5	5	5	The Good: The	The Bad: If y
stanfordu	August 1'	4	4	5	5	4	5	4	The Good: The	The Bad: The
stanfordu	June 11-	4	5	4	3	5	3	4	The Good: Sicn	The Bad: If y
stanfordu	June 11-	4	5	5	4	4	4	4	The Good: I dic	The Bad: I vi
stanfordu	June 23-	4	5	5	3	4	4	5	The Good: One	The Bad: Bay
stanfordu	July 27- 2	3	3	4	4	3	5	4	The Good: The	The Bad: A l
stanfordu	July 27- 2	5	5	5	5	5	5	5	The Good: The	The Bad: An
stanfordu	July 27- 2	4	5	5	5	5	5	4	The Good: I liv	The Bad: Ou
stanfordu	August 1'	4	5	5	4	5	5	5	The Good: Was	The Bad: I ha

Figure 4: *Data sample from* studentAdvisor.com.

3.2 Data Capturing & Retrieving Modules

Recently online social media has undoubtedly become the most popular and convenient means for the public to communicate, exchange opinions, and stay connected. Hence, studying online messages and formed online social networks has received a lot of attention from scholars and professionals worldwide. Sobkowicz et al. (2012) develop a framework using content analysis and sociophysical system modeling techniques, focusing on understanding and visualizing the formation of

political opinions and online networks over social media. Farina et al. (2014) present generally a practical, technical solution to extract and visualize massive public messages from different data sources.

To get the general understanding of public comments, we develop a sentiment analyzer to process public comments. The sentiment analyzer tries to answer two questions. First, it would like to determine the sentiment strength of a comment, i.e. quantifying how positive or negative a comment is. Secondly, it would like to understand in which perception areas a user tried to provide his/her comment, i.e. classifying comments.

To quantify how positive or negative a comment is, we extract relevant words based on a well-defined sentiment dictionary. The AFINN is a list of English words that is divided into positive and negative sentiments. Positive words ranges with the strength of from 1 to 5 and negative words ranges with the strength of from -1 to -5. The current version of AFFIN dictionary contains about 2500 words and phrases (Nielsen, 2011). Instead of using 10 levels of sentiment strength, we redefine the dictionary using 4 levels, defined as very positive (5 and 4), positive (3, 2, and 1), negative (-1, -2, and -3), and very negative (-4 and -5), focusing on finding the polarity of words in the text comments.

To support the categorization of words extracted from comments, the General Inquirer Dictionary (Stone, 1997) is then applied. The list of categories includes Academ (academic and intellectual fields), Coll (all human collectivities), Work (ways for doing work), SocRel (interpersonal processes), Place (place related words), Social (social interaction), Region (region related words), Exert (movement categories), and Quality (qualities or degrees of qualities). The occurrences of words labeled in the dictionary in a comment can be simply used as a sentiment indicator of the comment. Thus, the sentiment analyzer computes how many relevant words that appear in a comment. Using "The Good" and "The Bad" comments for Princeton University respectively, Figure 5 and 6 show sample results of sentiment analysis in this study. The method of validating dictionaries and relabeling words and detailed analysis in support of comments' categorization and classification are well presented in Ravi's thesis (2015).

sentence	vNeg	neg	pos	vPos	Academ	Coll	Work	SocRel	Place	Social	Region	Exert	Quality	pos
The Good: Princeton is the crean	0	1	4	1	1	0	3	1	2	1	0	1	0	positive
The Good: High quality of educat	0	0	3	0	4	2	1	1	1	1	0	1	1	positive
The Good: Princeton is one of th	0	0	3	0	1	1	0	0	2	1	0	0	0	positive
The Good: They have great educ	0	0	4	0	1	0	0	1	0	0	0	0	0	positive
The Good: The school is clean an	0	0	3	0	2	1	2	0	1	1	0	0	1	positive
The Good: One of the best colleg	0	0	3	0	0	0	0	0	0	0	0	0	0	positive
The Good: Princeton students ar	0	0	2	0	1	0	1	3	2	1	0	0	1	positive
The Good: The community was v	0	0	3	0	0	2	0	1	0	0	0	0	1	positive

Figure 5: An example of "The Good" comments.

sentence	vNeg	neg	pos	vPos	Academ	Coll	Work	SocRel	Place	Social	Region	Exert	Quality	neg
The Bad: Princeto	0	1	1	1	2	1	2	3	3	2	0	1	1	negative
The Bad: Traffic is	0	1	0	0	2	0	0	1	0	0	0	0	0	negative
The Bad: Princeto	0	4	2	0	5	1	1	2	3	3	0	0	4	negative
The Bad: The price	0	1	0	0	2	0	1	0	2	2	0	0	0	negative
The Bad: The mate	0	2	1	0	2	1	0	0	2	1	0	0	0	negative
The Bad: Tuition is	0	1	1	0	1	0	0	0	1	1	0	0	0	negative
The Bad: Princeto	0	1	0	0	0	0	0	0	2	1	0	1	1	negative
The Bad: I did not	0	1	1	0	0	0	0	0	0	0	0	0	0	negative

Figure 6: An example of "The Bad" comments.

Twitter.com is a very popular online social networking service, which essentially leverages real time push technologies and allows the public to post and read up to 140-characters microblogs called tweets. With the advent of smartphone technologies and services, mobile devices have made microblogging extremely handy and dynamic. Because of the vivid and pervasive, and short, easily understandable nature of microblogs, microblogging has substantially increased its popularity in the public (Yu and Qiu, 2014). Researchers start to pay much attention to understandings of tweets, aimed at getting better and real time understandings of various social behavior and market trends (Wu et al., 2010; Alper et al., 2011; Marcus et al., 2012).

16	massachu	Massachu	42.36364	71.09433	MIT	Researchers discover that aspartate is a limiter of #cellproliferation http://t.co/63MhyJDeOI @MIT
17	massachu	Massachu	42.36364	71.09433	MIT	#NewHorizons data hint at underground ocean http://t.co/Zr0YFsTIRS @MIT
18	massachu	Massachu	42.36364	71.09433	MIT	330 hp! Maximum torque of 410Nm! The #NewAstra speeds up for @TCRIntl! #TCRSeries http://t.co
19	massachu	Massachu	42.36364	71.09433	MIT	MIT looks to stay in vanguard of digital education: @MIT president Rafael Reif talks to @washingtor
20	massachu	Massachu	42.36364	71.09433	MIT	.@MIT is unveiling a $1.2 billion plan for Kendall Square http://t.co/xeQyTJurMb http://t.co/thcyIA(
21	harvardur	Harvard U	42.37791	71.11696	Harvard U	Mzee Jomo Kenyatta attended the London School of Economics while Barack Obama Snr went to Ha
22	princeton	Princeton	40.34479	74.65158	Princeton	Colleges with the top ROI: 1. Princeton University 2. Dartmouth College3. Williams CollegeMore: ht
23	princeton	Princeton	40.34479	74.65158	Princeton	#TopColleges 2015: 1. Pomona College2. Williams College3. Stanford UniversityMore: http://t.co/h:
24	princeton	Princeton	40.34479	74.65158	Princeton	St. Xavier OL Alex Deters committed to Princeton University today. http://t.co/I77tIwrlbP @stxspor
25	americanu	American	38.93706	77.0909	American	American University. FC Barcelona's first training session #tourFCB http://t.co/pHi0KtOgZM
26	americanu	American	38.93706	77.0909	American	'American 'Freshman': 12 Words That the University of New Hampshire Has Deemed 'Problematic' h
27	americanu	American	38.93706	77.0909	American	Second training session at the American University in Washington DC http://t.co/gVk0pNgsUF #Tou

Figure 7: An example of tweets retrieved from Twitter.com.

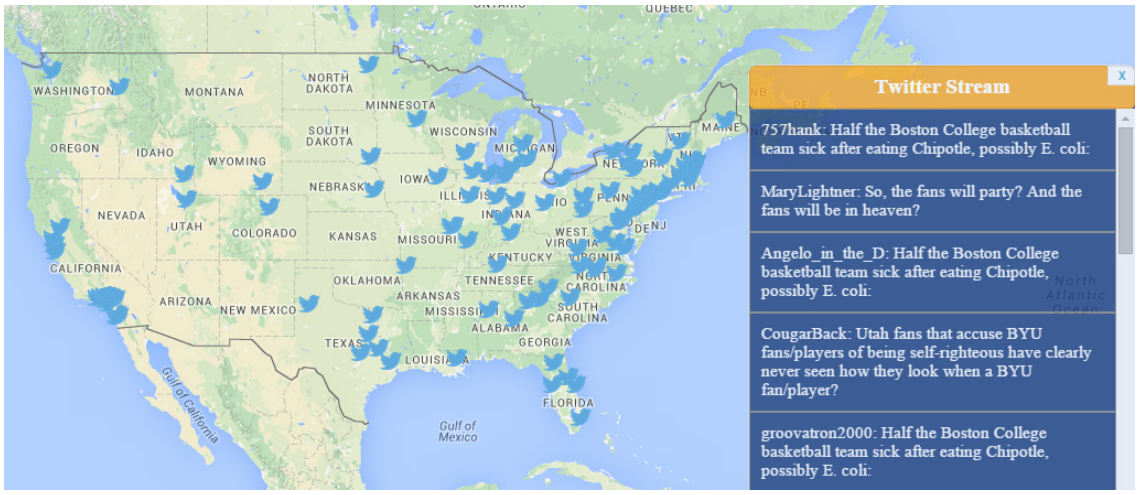


Figure 8: An interactive map view of sentiment trends of public opinions using streaming tweets.

To test out the concept of retrieving public opinions from a variety of data sources, tweets thus are used in this study. In this study, we focus on a list of targeted colleges in the US. Tweets from Twitter.com are retrieved using predefined queries (Figure 7) or streamed using a list of keywords in real time (Figure 8). Clean texts are extracted from received raw tweets and then further analyzed using our developed sentiment analyzer.

3.3 Big Data Computing Platforms

Data and information from the Internet are generally unstructured and mostly stored as images and texts. Our endeavor in enhancing service quality in this study substantially relies on the successful design, development, and deployment of big data computing platforms. Our deployed platforms use a scalable Lambda architecture to deal with big data volume and velocity simultaneously, supporting a hybrid computation model as both batch and real-time data processing can be combined transparently. The distribution layer consists of an Apache Kafka messaging broker. The batch layer includes HDFS, MapReduce, Hive, Pig, and Spark batch. The Apache Spark streaming layer includes Spark core and resilient distributed datasets, HBase, Cassandra, and MongoDB to perform lightning-fast cluster computing transformations and actions.

As shown in Figure 3, ratings, comments, and sentiment scores are processed in either batches or streaming, depending on how public opinions are retrieved from their data sources. Ratings, comments, and sentiment scores are aggregated, consolidated, and archived; they become readily available for use, i.e., visualizations or further aggregations and computations.

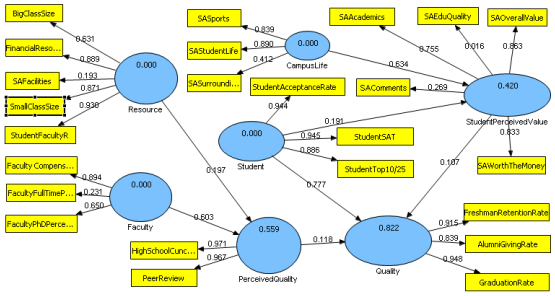


Figure 9: Enhancing the HESSEM by including sentiment scores of public opinions.

3.4 Ranking Modeler

Although demonstrating how educational service quality on the US higher education gets modeled in a quantitative and real-time manner is not the purpose of this paper, briefly showing how the assessment of service quality or performance gets enhanced by incorporating public opinions should be worthwhile. Figure 9 provides an overview of enhanced HESSEM models for top 100 colleges in the US, which has taken into consideration the above-mentioned sentiment scores computed from public opinions (Ravi, 2015; Huang & Qiu, 2015). A full list of new rankings can be found in Ravi (2015).

3.5 Visualizations of Aggregated Public Opinions and Sentiment Trends

As discussed in last section, the performance of educational services of a given college perceived by

Sentiment Analysis (last 24 hours)					
University	Positive Tweets Avg. SentiScore	Negative Tweets Avg. SentiScore	Positive Tweets Total SentiScore	Negative Tweets Total SentiScore	Net SentiScore
Massachusetts Institute Of Technology	0.44	-0.36	1,331.48	-507.14	824.34
Penn State University	0.47	-0.29	494.52	-164.64	329.87
University Of Miami	0.48	-0.12	298.73	-16.04	282.69
New York University	0.45	-0.24	349.48	-147.70	201.79
Duke University	0.68	-0.18	205.69	-5.97	199.72
Cornell University	0.31	-0.29	197.12	-7.54	189.58
The University Of Georgia	0.61	-0.45	205.11	-20.49	184.62
Johns Hopkins University	0.77	-0.32	187.21	-3.80	183.41
Columbia University	0.52	-0.32	184.70	-20.73	163.98
Brown University	0.53	-0.41	170.95	-65.21	105.75
Ball State University	0.48	-0.33	139.07	-47.46	91.62
The Ohio State University	0.50	-0.34	89.48	-16.43	73.05
University Of Florida	0.45	-0.30	83.29	-22.25	61.03
University Of Pennsylvania	0.52	-0.30	76.29	-15.34	60.95
University Of Maryland	0.19	-0.26	71.42	-10.79	60.63

Figure 10: A statistic report of sentiment trends of public opinions based on tweets.

the public changes with public opinions. If the proposed approach and system gets fully deployed with the ability of assessing the quality of college's services on a daily basis, aggregating and visualizing public opinions can then play an important role in helping stakeholders promptly understand what the public values the performance and quality of their provided education services. Sentiment trends can be one of effective indicators.

Sentiment trends could timely help administrators understand what the public is thinking about the moving direction of their provided services. A sudden jump of the number of tweets on a college might serve an alert, indicating that an event is currently drawing much public attention. Figure 8 shows an interactive map view of sentiment trends of public opinions using streaming tweets. By clicking a college tweet icon on the map, one can clearly see its sentiment trend for last 15 days or a customized interval. Figure 10 presents a statistic report of sentiment trends of public opinions based on tweets.

4 CONCLUSIONS

As discussed earlier, HESSEM allows ranking factors to be easily changed over time. But it is challenging to identify meaningful factors and then collect sufficient data for the identified factors. As public opinions play a key role in assessing customers' perception of their consumed services, this paper focused on introducing a systems approach to aggregating and visualizing public opinions. We demonstrated that capturing and understanding public ratings and comments on higher education helped enhance service quality assessment in general and develop a better and more effective rating system for education in the future.

By capturing and deciphering market trends in real time, the presented systems approach truly possesses promising potential of facilitating decision-making of addressing the needs of customers in the service industry. Although there will be a variety of research areas we could further our studies, collecting more data and information from other popular websites including facebook and Google trends and improving sentiment analysis accuracy in the education service domain are surely what we will work on in the near future. Through educational data mining and learning analytics, we could promptly uncover more insights to assist stakeholders in administrating and transforming their higher education practices in an effective and satisfactory manner. From a systems perspective, the proposed big data based evaluation system could become smarter and smarter as both the assessing model and the used service quality factors can be evolved over time.

ACKNOWLEDGEMENTS

This work was done with great support and help from the Big Data Lab at Penn State. Dr. Adrian Barb from Penn State significantly contributed to the deployment of big data platforms. The project entitled "*Big Data Platform (Massive Data) for Proactive Analyses of Behaviors of Users in Urban Worlds*" is financially supported by the Rhône-

Alpes Region, France. This project was also partially supported by IBM Grants (JLP201111006-1, 2011-12; IBM-NUAA-SUR, 2012-13: Customer Behaviour Analytics in Multi-channel Scenario) and the Penn State Faculty Development Research Fund (*Exploring Mechanisms for Enriching Mobile User Browsing Experience* – 2014-15; *Building a Foundation to Showcase Potential of an IoT Based "Sense and Respond" Framework* – 2015-16).

REFERENCES

Ahlquist, J., & Saagar, K. 2013. Comprehending the complete customer: leveraging behavioral and attitudinal data for actionable analytics. *Analytics*, May/June, 36-50.

Alper, B., Yang, H., Haber, E., & Kandogan, E. 2011. Opinionblocks: Visualizing consumer reviews. *IEEE VisWeek 2011 Workshop on Interactive Visual Text Analytics for Decision Making*.

Bergseth, B., Petocz, P., and Abrandt Dahlgren, M. 2014. Ranking quality in higher education: guiding or misleading? *Quality in Higher Education*, 20(3), 330-347.

Campbell, A.T., Eisenman, S.B., Lane, N.D., Miluzzo, E., Peterson, R.A., Lu, H., Zheng, X., Musolesi, M., Fodor, K. and Ahn, G.S., 2008. The rise of people-centric sensing. *IEEE Internet Computing*, 12(4), 12-21.

Farina, J., Mazuran, M., and Quintarelli, E. 2014. Extraction, Sentiment Analysis and Visualization of Massive Public Messages. *New Trends in Databases and Information Systems*, 159-168. Springer International Publishing.

Harvey, L. 2008. Rankings of higher education institutions: A critical review. *Quality in Higher Education*, 14(3), 187-207.

Huang, Z. and Qiu, R. G. 2016. A quantitative and mode-driven approach to assessing the US higher education. Accepted by *Quality in Higher Education*.

Labrecque, L. I., vor dem Esche, J., Mathwick, C., Novak, T. P., & Hofacker, C. F. 2013. Consumer power: Evolution in the digital age. *Journal of Interactive Marketing*, *27*(4), 257-269.

Marcus, A., Bernstein, M. S., Badar, O., Karger, D. R., Madden, S., and Miller, R. C. 2012. Processing and visualizing the data in tweets. *ACM SIGMOD Record*, 40(4), 21-27.

Meyer, C., & Schwager, A. 2007. Understanding customer experience. *Harvard Business Review*, 85(2), 116.

MIT., 2011. MIT Ranked 3rd in the World, 5th in the U.S.?. *MITFaculty Letter*, Retrieved Aug. 18, 2015 from http://web.mit.edu/fnl/volume/241/usnews.html.

Nielsen, F. Å. 2011. A new ANEW: Evaluation of a word list for sentiment analysis in microblogs. *arXiv preprint arXiv:1103.2903*.

Qiu, R. G. 2009. Computational thinking of service systems: dynamics and adaptiveness modeling. *Service Science*, 1(1), 42-55.

Qiu, R. G. 2014. *Service Science: the Foundations of Service Engineering and Management*, John Wiley & Sons: Hoboken, New Jersey, USA.

Qiu, R. G., Wang, K., Li, S., Dong, J. and Xie, M., 2014, June. Big data technologies in support of real time capturing and understanding of electric vehicle customers dynamics. *The 5th IEEE International Conf. on Software Eng. and Service Science*, 263-267.

Qiu, R. G., Ravi, R., and Qiu, L. 2015. Aggregating and visualizing publicati opinions and sentiment trends on the US higher education. *The 17th International Conference on Information Integration and Web-based Applications & Services*, 262-266.

Ravi, R. 2015. *A Quantitative and Big Data Driven Approach to Assessing the US Higher Education*. MS Thesis Paper, The Pennsylvania State University.

Sobkowicz, P., Kaschesky, M., and Bouchard, G. 2012. Opinion mining in social media: Modeling, simulating, and forecasting political opinions in the web. *Government Information Quarterly*, 29(4), 470-479.

Stone, P. J. 1997. Thematic text analysis: New agendas for analyzing text content. *Text Analysis for the Social Sciences. Mahwah, NJ: Lawrence Erlbaum*, 33-54.

Wu, Y., Wei, F., Liu, S., Au, N., Cui, W., Zhou, H., & Qu, H. 2010. OpinionSeer: interactive visualization of hotel customer feedback. *IEEE Transactions on Visualization and Computer Graphics*, *16*(6), 1109-1118.

Yu, Y., and Qiu, R. G. 2014. Followee recommendation in microblog using matrix factorization model with structural regularization. *The Scientific World Journal*, 1-10.

Dealing with the Complexity of Model Driven Development with Naked Objects and Domain-Driven Design

Samuel Alves Soares, Mariela Inés Cortés and Marcius Gomes Brandão
State University of Ceará, Dr. Silas Munguba Avenue, 1700, Fortaleza, Brazil
samuel.soares@uece.br, mariela@larces.uece.br, marcius.brandao@uece.br

Keywords: Model-Driven Development, Naked Objects, Domain-Driven Design, Domain Patterns, Design Patterns.

Abstract: The Model-Driven Development aims to the implementation of systems from high-level modeling artifacts, while maintaining the focus of the development team in the application domain. However, the required models in this approach become very complex and, in many cases, the developer's intervention can be required along the application infrastructure construction, then failing to keep the focus on application domain and could also be impaired synchronization between code and model. To solve this problem, we propose a tool where the developer just models the business objects through the use of Domain Patterns and Software Design Patterns, which is used to generate the application code. A naked object framework is responsible for the system infrastructure code. The use of the tool benefits the generation of functional applications, while maintaining the synchronization between code and model along the development.

1 INTRODUCTION

Throughout its evolution, the software engineering has looking for to abstracting increasingly the developing work from the computing infrastructure (Hailpern and Tarr, 2006; Thomas, 2004). The full focus on the problem domain has been claimed as the ideal model for the computer systems development (Pawson, 2004; Budgen, 2003). In this sense, in the Model-Driven Development (MDD), the design models are used as primary artifacts in system development, going beyond to the specification and design phases (Brambilla et al., 2012; Mohagheghi and Aagedal, 2007).

However, the construction of a complete software using the MDD approach requires the definition of infrastructure aspects, such as user interface (UI) and persistence technologies, both in the model or code generated, by taking the focus of the application domain (Hailpern and Tarr, 2006). As consequence, it makes the modeling more complex and less intelligible since several artifacts from a specific platform must be included (Hailpern and Tarr, 2006; Thomas, 2004). In addition, the ambiguous nature of models and the redundancy of the information along the different visions, makes it difficult to maintenance and impairs the adoption of MDD in the industry (Haan, 2008; Hailpern and Tarr, 2006). In order to solve these questions, complementary approaches must be considered (Whittle et al., 2013).

In the context of object-oriented development, the Naked Objects Pattern (NOP) (Pawson, 2004) promotes focus on the implementation of the domain objects. Meanwhile, a framework is responsible to generate all the system infrastructure automatically. Thus, is possible to create an application based only on the implementation of the domain objects avoiding redundancy and replicated information. In other hand, objects in the domain model can be documented on the basis of the Domain-Driven Design (DDD) approach (Evans, 2003).

Considering solutions centered on the application domain, research show the suitability of NOP in the context of DDD approach for the development of robust systems (Haywood, 2009; Läufer, 2008). Likewise, the utilization of design patterns in association with DDD features can increase the productivity of the application development and maintenance (Nilsson, 2006; Fowler et al., 2003; Gamma et al., 1995). This association contributes in the identification of the responsibility of each class in the application model, in order to facilitate understanding of the model and their corresponding implementation code (Nilsson, 2006).

Thus, in view of the problematic of the model-driven development and the solutions for code generation and modeling centered on the application domain we propose a MDD solution whose modeling

Soares, S., Cortés, M. and Brandão, M.
Dealing with the Complexity of Model Driven Development with Naked Objects and Domain-Driven Design.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 528-535
ISBN: 978-989-758-187-8

is based on DDD and software patterns and the full code of the application domain is generated based on the NOP to run.

2 THEORETIC REFERENTIAL

2.1 Model-Driven Development

Model-Driven Development (MDD) is a development methodology that foresees the generation of executable code starting from high-level models, or even model execution, enabling developers to work in higher abstraction level. It promotes the rapid development of applications and facilitates the communication among the project members (Hailpern and Tarr, 2006; Brambilla et al., 2012).

In the counterpart, a useful model artifact in MDD must be sufficient to execute or require a minimum intervention to transform it into executable code. Thus, a complete modeling of the system is required, including details about the presentation technologies, for example (Brambilla et al., 2012). It makes the modeling laborious and result in large and more complex designs (Hailpern and Tarr, 2006).

The utilization of standards languages such as the Unified Modeling Language (UML), provides a uniform notation and favors their utilization for a wide range of activities. But in return, it becomes huge, ambiguous semantic and unwieldy, with redundant information along the diverse diagrams. Thus, keeping the synchronization and consistency between them, avoiding information loss, is hard and hinder the use of MDD in the industry (Haan, 2008; Hailpern and Tarr, 2006; Thomas, 2004).

In this sense, the use of patterns in the system modeling enriches the semantics without increasing complexity to the model, contributing with the software maintenance (Evans, 2003).

2.2 Naked Objects Pattern

The Naked Objects Pattern (NOP) focuses on the creation of domain objects to their direct presentation to the user (Pawson, 2004). The pattern states that the infrastructure aspects, such as presentation, persistence and remote communication, must be supplied by a framework (Haywood, 2009; Pawson, 2004). In this way the software developer is responsible only for the creation of the domain classes and their relationships, states and behaviors.

In practice, the creation of a new class in the NOP presupposes its modeling in terms of attributes and methods, for example using UML notation (Booch et al., 2006). Using a suitable template for the code generation, the application can be executed through of framework based on NOP (Läufer, 2008). In this sense, this solution based on NOP becomes adequate to the problematic pointed in MDD.

However, there are objects in the model need to be identified as persistent objects or as simply attributes of other objects, for example. In this sense, DDD approach and design patterns can be useful.

2.3 Domain-Driven Design

The Domain-Driven Design (DDD) (Evans, 2003) is a set of principles, techniques and patterns for software development. The focus of the DDD is the domain, abstracting infrastructure aspects.

In this context, Domain Patterns aims to identify the characteristics and responsibilities of each domain objects in the application in order to create the Domain Model (Evans, 2003). This identification can be performed by UML stereotypes (Booch et al., 2006) or colors (Coad et al., 1999). The main are:

- *Entity*: an object that maintains continuity, it has an identity, and it has a life cycle;
- *Value Object* (VO): an object used to describe other objects and has no identity concept;
- *Service*: a class that provides services to objects and without keeping a state;
- *Aggregate*: it represents related Entities and VOs that are treated as a unit. It has a root object;
- *Repository*: a mechanism to the insertion, removal and queering of objects abstracting the database.

There are studies that show the usefulness of DDD approach with NOP in the creation of robust systems (Haywood, 2009; Läufer, 2008). In this context, the creation of an application requires the construction of a Domain Model indicating the Domain Patterns associated with the classes of the application. In addition, design patterns (Gamma et al., 1995; Fowler et al., 2003; Nilsson, 2006) can be used in association with Domain Patterns in order to solve common development problems (Nilsson, 2006).

2.4 Design Patterns

Design Patterns are reusable solutions to recurring problems in the object-oriented software design (Gamma et al., 1995), and they are an excellent tool to express the concepts involved in a particular domain (Buschmann et al., 2007). The design patterns are divided into three categories: Creational, Structural and Behavioral (Gamma et al., 1995).

Design patterns can be used together with Domain Patterns to refine the domain model (Nilsson, 2006) and assist the identification of the responsibility of each class in the application, in order to facilitate the model of understanding and generation of the appropriated code (Läufer, 2008). For example, the State pattern can be useful to express the various states associated with an Entity class.

2.5 Patterns of Enterprise Application Architecture

The Patterns of Enterprise Application Architecture (PoEAA) (Fowler et al., 2003) were caught along the development of enterprise object-oriented systems. These patterns are used to drive the code generation.

The Identity Field pattern, for example, is associated with an Entity class to link the Entity to a table in the database. Moreover, the Aggregate is directly related to the Encapsulate Collection pattern which ensures the control and consistency between items and the root object. Thus the necessary methods in the Encapsulate Collection pattern can be generated automatically. The concurrency control is treated with the Coarse-Grained Lock pattern. Finally, the Business ID pattern (Nilsson, 2006) identifies properties that are business keys of object and that ensure the uniqueness of object.

2.6 UI Conceptual Patterns

Despite the possibility of taking the whole application just creating domain objects and be able to run the application without to model the infrastructure code, the NOP can generate only one UI (Pawson, 2004).

UI Conceptual Patterns (Molina et al., 2002b) can be used to specify UI for independent devices. These interfaces can be refined using UI Design Patterns, as well as be used to automatically obtain specific UI prototypes for various devices. These patterns are composed of simple patterns and they are categorized into four types, namely: Service Presentation, Instance Presentation, Population Presentation and Master-Details Presentation (Molina et al., 2002a).

Through these patterns the developer can customize the view of the objects to the user via multiple visions without having to deal directly with UI infrastructure code. The Naked Objects View Language (NOVL) (Brandão et al., 2012a) allows the framework based on NOP manages various visions for the same object based on the UI Conceptual Patterns.

3 RELATED WORK

In general, the modeling tools work with visual modeling, such as UML, in order to help in the software development activities. Many seek to support MDA (Kleppe et al., 2003) as the creation of platform independent models and subsequent code generation in an object-oriented programming language. Examples of tools with these characteristics are: Enterprise Architect (EA)[1], Modelio[2], Objecteering[3], and objectiF[4], Such modeling tools support code generation, however little or no support is provided for the generation of infrastructure code.

Some of these tools support the creation of templates to support the automatic generation of the infrastructure code. However, synchronization problems can arise and manual alterations can be required. Another limitation relates to relationships between diagrams, for example as denote the association between a dynamic diagram that modeling the behavior of a class's operation.

The lack of a tool to support the development of the application model infrastructure in integrated way, turns the development using the MDD approach complex, leading to possible incompatibilities between the tools used in the process (Alford, 2013). On the other hand, MDD projects focused on mechanisms to support model to model and model to code transformations, considering as a starting point models created from different modeling tools. Examples of these projects are AndroMDA[5], BaseGen[6], Jamda[7] and openMDX[8].

With creation of the complete models, MDD frameworks are able to create the business classes and infrastructure of the complete system. However, after that, any change in the model may need for manual intervention to avoid the overwritten of existent code. In addition, considering that the generation of the application is often based on layered architecture, changes to the domain layer can lead to alterations in other layers (Pawson, 2004).

So, any of the tools or framework cited above works in a satisfactory way in order to abstract

[1]EA - http://www.sparxsystems.com.au/products/ea/

[2]Modeliosoft - https://www.modelio.org/index.php

[3]Objecteering - http://www.objecteering.com/

[4]microTOOL objectiF - http://www.microtool.de/en/objectif-model-driven-development/

[5]AndroMDA.org - http://www.andromda.org/

[6]BaseGen - http://sourceforge.net/projects/basegen/

[7]Jamda Project - http://jamda.sourceforge.net/

[8]openMDX - http://sourceforge.net/p/openmdx/wiki/Introduction/

infrastructure aspects of the application without generating models and redundant code and an integrated manner.

4 THE Elihu MDD TOOL

Elihu is an MDD tool dedicated to the development of enterprise applications through the implementation of business domain objects. It is based on the concepts and patterns of DDD, software design patterns and NOP. Its aim is to create platform-independent models that contain all the functionality of the application on domain objects, and regardless aspects of infrastructure. The generation of user interfaces, persistence, security, among other things are possible through the NOP.

In Elihu, as shown in Figure 1, the Domain Model is created from the DDD Domain Patterns and software patterns. The Domain Patterns represent the application building blocks (Evans, 2003) and they are associated with software patterns to allow the representation of all the features, operations and visions of domain objects (Nilsson, 2006). After modeling the domain objects, *templates* are applied to generate the source code according to the desired language and platform. This source code is then submitted to a framework based on NOP to run.

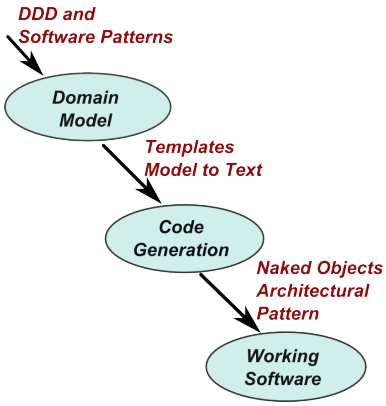


Figure 1: Development process using Elihu.

4.1 Elihu's Metamodel

Elihu's metamodel (Figure 2) defines the *DomainModel* metaclass to represent the application model. Domain Patterns are defined by metaclasses, i.e., *Aggregate, Entity, Service* and *ValueObject*. The metaclasses are linked to the *DomainModel* and used as the application modeling elements. The relationships between model elements are defined by the *Association* metaclass.

The *Classifier* metaclass is designed to define the common characteristics of *Entity* and *VO* in an inheritance relationship. *Classifiers* have properties, operations, and may be part of associations.

The *Aggregate* metaclass defines a set of *Classifiers* that behave as one logical unit. There is a root, which is necessarily an *Entity*. The *Service* metaclass defines an element that provides operations and does not have state.

A *Property* metaclass needs to be properly configured when added to a *Classifier*, so it can be interpreted correctly when the application generating. The main *Property*'s attributes are: *name*, *type*, *scale*, *length*, *required*, *visibility*, *minValue*, *maxValue*, *transient*, *mask*, *readOnly*, *lower* and *upper*. Regarding an *Operation*, its main attributes are: *name*, *return*, *body* and *visibility*.

The value of the *body* attribute of the Operation metaclass can be informed in textual form or through behavioral diagrams. *Operation* may also have input parameters, as normally happens in object-oriented languages. Thus, *Operation* metaclass is associated to *Parameter* metaclass in the metamodel, which in turn inherits from *Property* metaclass. When the developer defines an operation's *Parameter*, he should set of the *Parameter* properties, similarly as *Property*.

The metamodel also defines relationships between *Classifier* and *Aggregate* metaclasses and software patterns. The main patterns supported are:

- *Business ID* (Nilsson, 2006) - it is the identification of properties that are *Entity*'s business keys and that guarantee its uniqueness;
- *Presentation* (Molina et al., 2002a) - it is the UI definition of *Classifier* or *Aggregate*. It is represented by metaclasses which contains attributes corresponding to properties defined to generate UI, in this case using NOVL;
- *Specification* (Evans, 2003) - it is the definition of conceptual specifications of an object, such as queries based on domain concepts, which can be reused;
- *State* (Gamma et al., 1995) - it is the representation of the states in which an object goes through during its life cycle. In this case, a state diagram binding to the *Entity* metaclass defines the states and transitions of the object.

This metamodel has been implemented with the *Ecore* metamodel language, which is part of *Eclipse Modeling Framework* (EMF)[9]. It is used to create model application and the modeling information are used to generate a XMI file (Brambilla et al., 2012).

[9]EMF - https://www.eclipse.org/modeling/emf/

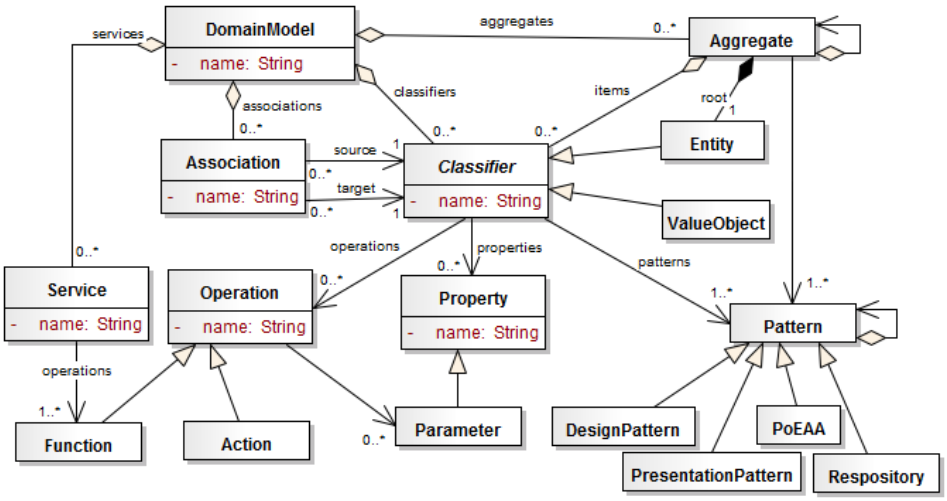


Figure 2: Elihu's Metamodel.

This file is used in the process of code generation by *templates* presented in the next section.

4.2 Elihu's Code Generation Templates

The Elihu includes *templates* to support the code generation of the modeled objects according to the characteristics and composition of each pattern. These *templates* have been created through the Eclipse plugin Acceleo[10]. The code generation occurs from XMI file of the model created by the developer.

The *templates* generate code in Java programming language in the structure of *Entities Framework* that implements the NOP (Brandão et al., 2012b). Other *templates* can be created and added to Elihu to generate code for other *frameworks* based on NOP.

The code snippet below refers to the *generateEntityPattern template*. This template defines the rules of code generation from a *Entity*:

```
[template public
   generateEntityPattern(entity : Entity)]
[file (entity.name.toUpperFirst()
  .concat('.java'), false)]
       ... package and imports
@Entity
[if (not entity.businessId -> isEmpty())]
   //Business ID Pattern
   @Table(uniqueConstraints =
     {@UniqueConstraint(columnNames =
      {[writeUniqueConstraints(entity)/]})})
[/if]
[if (entity.presentations -> size() > 0)]
  //Presentation Pattern
  @Views({[for (s : Presentation |
   entity.presentations) separator(',\n')]
   @View(name = "[s.name/]",
    title = "[s.title/]",
    //Filter Pattern
    filters = "[s.filters/]",
    //Display Set Pattern
    members = "[s.displaySet/]",
    //Specification Pattern
    namedQuery=
      "[s.specification.definition/]",
    template="[s.template/]")[/for]})
[/if]
public class [entity.name.toUpperFirst()/]
   implements Serializable {
       ...
  [if(not entity.state.oclIsUndefined())]
  //State Pattern
  [for(t : StateTransition | entity.state
       .transitions) separator('\n')]
   public String [t.name/]() {
     [entity.getStateEnum()
      .toLowerFirst()/].[t.name/](this);
     return "[t.target.name/][entity.name/]";
   } [/for] [/if] ...
} [/file] [/template]
```

[10]Acceleo - https://eclipse.org/acceleo/

For each *Entity* in the model, the template above creates a *.java* file for the class, checks which patterns are linked to class, and creates the structure of a Java class with the annotation *@Entity* of Java Persistence API (JPA) (Jendrock et al., 2014) to ensure the persistence of objects of that class.

If the *Business ID* pattern is linked to the *Entity* the template adds the *UniqueConstraints* annotation and configures class business keys and it creates the *hashCode* and *equals* methods based on these business keys (Jendrock et al., 2014). If the *Presentation* pattern is linked, the template adds *@Views* and *@View* annotations of Entities Framework to define UIs with NOVL. Each item

in the *presentations* collection represents a UI. If the *Specification* pattern is linked, the template adds *@NamedQueries* and *@NamedQuery* annotations of JPA to query specification of the class (Jendrock et al., 2014). If the *State* pattern is linked, the template creates the class structure for the possible states and the methods that perform switching entity state in accordance with the transition rules.

As shown in Section 2.5, there are other patterns that may be related to the Domain Patterns as *Identity Field*, *Encapsulate Collection* and *Coarse-Grained Lock* (Fowler et al., 2003). These patterns do not need to be added explicitly by the developer in modeling. They can be automatically generated according to the characteristic of the modeled object. The code snippet below shows these cases:

```
[template public
   generateEntityPattern(entity : Entity)]
       ...
public class [entity.name.toUpperFirst()/]
   implements Serializable {
  //Identity Field Pattern
  @Id @GeneratedValue private Long id;
       ...
  [if (entity.aggregateItem
       .oclIsUndefined())]
  //Coarse-Grained Lock Pattern
  @Version private Timestamp version; [/if]
      ...
  [if (not entity.aggregate
       .oclIsUndefined())]
  //Encapsulate Collection Pattern
  [for (a : Association | entity.outgoing)]
   public void add[a.target.name/]() {
       [a.target.name/] item =
           new [a.target.name/]();
       item.set[entity.name/](this);
       [a.targetDef.name/].add(item);
       numberOf[a.targetDef.name/]++;
   } [/if] [/for]
   [/if]
   [for (a : Association | entity.incoming)]
       [if (not entity.aggregateItem
           .oclIsUndefined() and entity
           .aggregateItem.root = a.source)]
   public void remove[a.target.name/]() {
     [a.sourceDef.name/].get[a.targetDef
      .name.toUpperFirst()/]().remove(this);
     [a.sourceDef.name/].setNumberOf[a
      .targetDef.name/]([a.sourceDef.name/]
      .getNumberOf[a.targetDef.name/]()-1);
   } [/if] [/for]  ...
} ... [/template]
```

For all *Entity* a *id* property is created, to bind the entity to a line in the corresponding table in the database, and is created a property with the annotation *@Version* for concurrency treatment (Jendrock et al., 2014). If the entity is the root of a *Aggregate*, access to other elements of aggregation should be controlled by that entity by the *add()* and *remove()* methods and other properties of control.

Finally, the template generates the properties, associations and operations of the class. The template checks the attributes configured by the developer to the correct mapping of code. The generation of operations sets the parameters set by the developer and the method body.

4.3 Example of Operation

In this section, the Elihu metamodel is instantiated to illustrate its operation.

This application consists of creating an order to sell products to customers with available credit limit in the company. The client must be registered with the identification number, the social identification number, name and address. The customer's credit limit should consider orders unpaid customer. Each order, in turn, must have the date of creation, a number and the items for sale with product identification, quantity of items and value. Must be identified if a order has been accepted, canceled or has been paid. It should also be possible to consult all orders placed by the customer in a specific data. Figure 3 shows the model created based on these requirements.

For the application has been created *Customer, Order, Product* and *OrderLine Entities*, *Address VO*, associations between *Customer* and *Address*, *Order* and *Customer*, *Order* and *OrderLine* and *OrderLine* and *Product*, and *TotalCreditService Service* with *getCurrentCredit* method to provide total available to a customer credit. Also added the properties of the *Entities* and the methods of *Customer* and *Order*. *Order* and *OrderLine* form an *Aggregate* where *Order* is the root.

As example, Figure 4 shows the setting details of the *number* property of Order. *Number* has been set as String of ten characters, required and has only one value. These characteristics are used to generate code, database and UI, avoiding duplication of validations and settings by the developer. The necessary changes in properties are held only at this location and it is reflected in other points where it is used without the developer needs to do manual changes.

In the *Order Entity* has been also added three *Presentations* to different user profiles. Figure 5 shows the details of the *Presentation ListOfOrders*, which defines the UI regarding query and presentation of orders. The *Filter* and *Display Set* have been added under the rules of NOVL language, and set

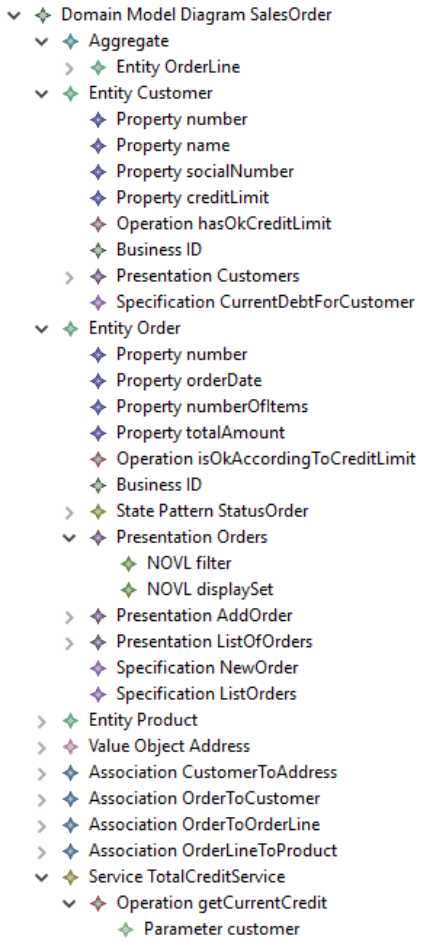


Figure 3: Sales order application.

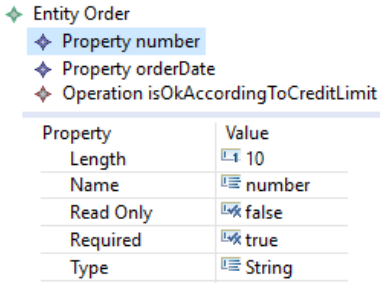


Figure 4: Order's *number* property.

the *Specification* of how the query will be held is the ListOrders. *Presentation Orders* defines a UI where the user can view and add orders and *Presentation AddOrder* defines a UI to creating orders.

A state machine for the *Order Entity* called *StatusOrder* has been also created. The states and transitions are shown in Figure 6, being highlighted the *accept* transition, as example, which defines *New* state as source and *Accepted* state as target. The other transitions are *pay* of *Accepted* to *Paid* and *reject* of

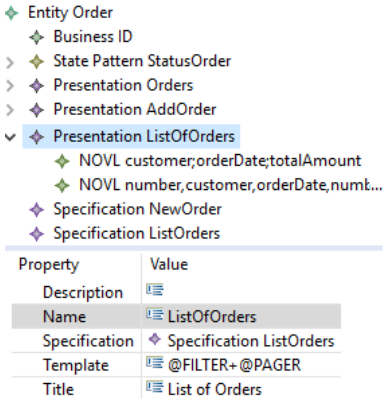


Figure 5: *Presentations* of *Order Entity*.

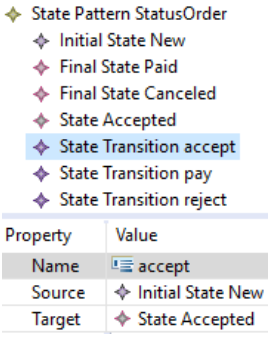


Figure 6: States of *Order*.

Accepted to *Canceled*.

With this complete model created, the application code can be generated (with patterns, states, bahaviors, and constraints) and executed. Figure 7 shows *Presentation Orders* presented to the user. Figure 7 shows the UI after the user has added two items to the order and has used the *Accept* operation.

Necessary changes in application because of changing requirements or because maintenance must be performed in the model.

5 CONCLUSION

The software modeling in the context of MDD need to consider infrastructure aspects during the creation of classes to generate complete models useful for the generation of functional software. Consequently, models became most complex and takes away the developer's focus of the application domain

This work presented Elihu, a MDD tool that includes the utilization of NOP, Domain Patterns and Design Patterns to create complete models abstracting the application infrastructure. Thus, developers can only be concerned to the application domain, reducing the complexity in system modeling.

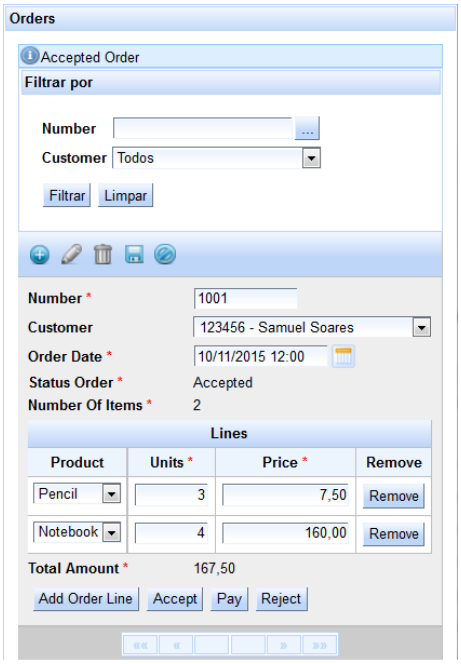


Figure 7: Orders UI.

Elihu generates executable applications through modeling of application domain objects. The model presents clarity about the purpose of the system due to the use of patterns. Additionally, the generated code is also reliable to the system domain and the developer does not need to change infrastructure code. In addition, it can change the domain model, due to changing requirements, and synchronize automatically with code.

As future work, we propose the creation of concrete notation to support the visual modeling; creating of nested aggregates; automatic detection of the structure of patterns *State* and *Encapsulate Collection*; the inclusion of new patterns in the metamodel, and the creation of *templates* to others NOP *frameworks* that serve different platforms.

REFERENCES

Alford, R. (2013). An evaluation of model driven architecture (mda) tools. Master's thesis, University of North Carolina Wilmington, Wilmington, NC.

Booch, G., Rumbaugh, J., and Jacobson, I. (2006). *UML: guia do usuário*. Campus, Rio de Janeiro.

Brambilla, M., Cabot, J., and Wimmer, M. (2012). *Model-driven software engineering in practice*. Morgan & Claypool Publishers.

Brandão, M., Cortés, M., and Gonçalves, E. J. T. (2012a). Naked objects view language.

Brandão, M., Cortés, M. I., and Gonçalves, E. J. T. (2012b). Entities: A framework based on naked objects for development of transient web transientes. In *CLEI-Latin American Symposium on Software Engineering Technical, Medellim*, volume 4.

Budgen, D. (2003). *Software design*. Pearson Education, 2 edition.

Buschmann, F., Henney, K., and Schmidt, D. C. (2007). *Pattern-Oriented Software Architecture: On Patterns and Pattern Languages*, volume 5. John Wiley & Sons, Chichester.

Coad, P., Luca, J. d., and Lefebvre, E. (1999). *Java modeling in color with UML: Enterprise Components and Process*. Prentice Hall.

Evans, E. (2003). *Domain-Driven Design: tackling complexity in the heart of software*. Addison Wesley, Boston.

Fowler, M., Rice, D., Foemmel, M., Hieatt, E., Mee, R., and Stafford, R. (2003). *Patterns of enterprise application architecture*. Addison-Wesley Professional, Boston.

Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1995). *Design patterns: elements of reusable object-oriented software*. Addison Wesley, Indianapolis.

Haan, J. D. (2008). 8 reasons why model-driven approaches (will) fail.

Hailpern, B. and Tarr, P. (2006). Model-driven development: The good, the bad, and the ugly. *IBM systems journal*, 45(3):451–461.

Haywood, D. (2009). *Domain-driven design using naked objects*. Pragmatic Bookshelf.

Jendrock, E., Cervera-Navarro, R., Evans, I., Haase, K., and Markito, W. (2014). *The Java EE 7 tutorial*. ORACLE.

Kleppe, A. G., Warmer, J. B., and Bast, W. (2003). *MDA explained: the model driven architecture: practice and promise*. Addison-Wesley Professional.

Läufer, K. (2008). A stroll through domain-driven development with naked objects. *Computing in Science and Engineering*, 10(3):76–83.

Mohagheghi, P. and Aagedal, J. (2007). Evaluating quality in model-driven engineering. In *Proceedings of the International Workshop on Modeling in Software Engineering*, MISE '07, pages 6–, Washington, DC, USA. IEEE Computer Society.

Molina, P. J., Meliá, S., and Pastor, O. (2002a). Just-ui: A user interface specification model. In *Computer-Aided Design of User Interfaces III*, pages 63–74. Springer.

Molina, P. J., Meliá, S., and Pastor, O. (2002b). User interface conceptual patterns. In *Interactive Systems: Design, Specification, and Verification*, pages 159–172. Springer.

Nilsson, J. (2006). *Applying Domain-Driven Design and patterns - with examples in C# and .NET*. Addison Wesley Professional.

Pawson, R. (2004). *Naked Objects*. PhD thesis, Trinity College, Dublin.

Thomas, D. (2004). Mda: Revenge of the modelers or uml utopia? *Software, IEEE*, 21(3):15–17.

Whittle, J., Hutchinson, J., Rouncefield, M., Burden, H., and Heldal, R. (2013). Industrial adoption of model-driven engineering: are the tools really the problem? In *Model-Driven Engineering Languages and Systems*, pages 1–17. Springer.

MAS Ontology: Ontology for Multiagent Systems

Felipe Cordeiro[1], Vera Maria B. Werneck[1], Neide dos Santos[1] and Luiz Marcio Cysneiros[2]
[1]*Universidade do Estado do Rio de Janeiro, Master Program in Computational Science, Rio de Janeiro, Brazil*
[2]*York University, School of Information Technology, Toronto, Canada*
{vera, neide}@ime.uerj.br, fpaula@inf.puc-rio.br, cysneiro@yorku.ca

Keywords: Agent-based Software Engineering, Domain Ontology, Comparison of Agent-oriented Methodologies.

Abstract: This work describes the Multiagent Systems (MAS) Ontology to assist in the development of multi-agent system using different methodologies. The MAS Ontology consists of fragmenting agent-oriented methodologies following an ontology approach based on the best aspects of four prominent AOSE methodologies and Guardian Angel exemplar that identify the strengths, weaknesses, commonalities and differences. In this paper, we present a brief explanation of Multiagent methodologies and the step-by-step process to describe the agent-based systems domain and how it can be represented. Given the numerous works in the literature about MAS methodologies, our aim is to help select the best and more appropriate properties to be used in Multiagent Systems development.

1 INTRODUCTION

In the past two decades, the agent technology approach has been considered as a new paradigm for developing complex systems. This approach has attracted an increasing amount of interest from the research community and has demonstrated its potential in many fields, such as: (i) working with different types of distributed devices (e.g., sensor networks, mobile phones, and personal computers), (ii) enabling various types of communication and data exchange (e.g., audio and video), and (iii) ability to dynamically adapt to the ever changing requirements and dynamic operating environment (Munroe et al., 2006), (Pěchouček and Mařík, 2008), (Dam and Winikoff, 2013).

Agent-oriented systems must be built in terms of autonomous task-oriented entities. They need to be organized to interact (cooperate, coordinate and negotiate) with one another. To adopt the agents' perspective requires a new set of tools to support software development (Cernuzzi, Cossentino and Zambonelli, 2005).

Currently, we are faced with a multitude of different frameworks, some of them even supported by tools. However, very few methodologies are broad enough to support the whole software development life cycle or to support the complexity of developing such systems. Years ago, Luck, Mcburney and Preist (2003) stated: "One of the most fundamental obstacles to large-scale take-up of agent technology is the lack of mature software development methodologies for agent-based system".

In this work, we assume "methodology" as a set of phases that a practitioner must go through to design an agent-based system. We see a methodology as being composed of general concepts (deals with the question of whether a methodology adheres to the basic notions of agents and multiagent systems), specific concepts (underlying one particular capability or a characteristic), notation (symbols used to represent elements), modeling techniques (set of models that depict a system at different levels of abstraction and different aspects of the system), process (development aspect) and pragmatics (practical implementation aspects) (Sturm and Shehory, 2004).

The main goal of this paper is to provide an ontology structure for selecting the best and more appropriate artefacts to be used to develop one particular Multiagent Systems, the MAS Ontology. It is motivated by a large number of existing approaches and supported by our experience in using some of them. It is important to notice that this work does not claim nor intends to be complete. It is expected to be a first approach that will be perfected overtime but that will yet be of importance to help developers to use the best each of the current four (Gaia, MaSE, Prometheus and Tropos) methodologies covered in this work has to offer.

The main goal of the ontology proposed in this work is to capture and facilitate the reuse of

Cordeiro, F., Werneck, V., Santos, N. and Cysneiros, L.
MAS Ontology: Ontology for Multiagent Systems.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 536-543
ISBN: 978-989-758-187-8

knowledge gained through the evaluation of several MAS methodologies based on more than 20 projects developed with different methodologies using Guardian Angel (GA) Exemplar proposed by Yu and Cysneiros. However, to complement the ontology we added experiences extracted from other well know evaluation studies from the literature. We populated the ontology with the four methodologies because they were evaluated by GA, Sturm and Shehory (2004, 2014) and Dam and Winikoff (2004, 2013).

2 AGENT-ORIENTED METHODOLOGIES

Cernuzzi et.al. (2005) suggests a clean and disciplined approach to analyzing, designing and developing multiagent systems, using specific methodologies and techniques by means of notations, diagrams and tools to support the development.

We assume that each method has strengths and weakness, and these characteristics may influence the use of one methodology over another for one specific project. To validate the MAS Ontology we used four methodologies, namely, Gaia (Zambonelli, Jennings and Wooldridge, 2003), (Wooldridge, Jennings and Kinny, 2000), MaSE (Deloach, 2001), (Deloach, 2004), Prometheus (Padgham and Winikoff, 2002), (Padgham and Winikoff, 2003) and Tropos (Bresciani et al, 2004).

Jennings and Wooldridge proposed Gaia in 1999. It was extended and modified by Zambonelli in 2000 (Wooldridge, Jennings and Kinny, 2000), finally Zambonelli, Jennings and Wooldridge presented a stable version in 2003 (Zambonelli, Jennings and Wooldridge, 2003). Unlike many other methodologies, Gaia starts from modelling requirements. Later it guides developers to a well-defined design for the multiagent system, that way programmers can easily model and implement it, while dealing with the characteristics of complex and open multiagent systems.

MaSE methodology is heavily based on UML and RUP. It is divided into seven phases: capturing goals, applying use cases, refining roles, creating agent classes, constructing conversations, assembling agent classes and system design (Deloach, 2001), (Deloach, 2004).

Prometheus is an iterative methodology covering the complete software engineering process while aiming at the development of intelligent agents (in particular BDI agents). The concepts applied are goals, beliefs, plans, and events, resulting in a specification that can be implemented with JACK (Coburn, 2000). Prometheus covers three phases: the system specification, architectural design phase, detailed design phase (Padgham and Winikoff, 2002), (Padgham and Winikoff, 2003).

Tropos relies on the notion that an agent is based on goals and tasks adopted by the i* framework (Yu, 2009) and offers supports to applications, particularly for the development of BDI agents and the agent platform JACK. (Coburn, 2000). Tropos consists of four phases: early requirements, late requirements, architecture design, detailed design and implementation (Bresciani et al, 2004), (Tropos, 2014), (Coburn, 2000).

3 EVALUATION OF AGENT ORIENTED METHODOLOGIES

Several evaluations of agent orientated methodologies have been published (Dam and Winikoff, 2014), (Sturm and Shehory, 2014), (Dam, 2003), (Dam and Winikoff, 2004), (Tran and Low, 2005), (Elamy and Far, 2008), (Iglesias, Garijo and González, 1999), (Cernuzzi, Rossi and Plata, 2002), (Sure, Staab and Studer, 2002). Sturm and Shehory (2004, 2014), and Dam and Winikoff (2004, 2013), (Dam, 2003) were the most cited works in the MAS area.

Sturm and Shehory (2004), proposed a framework for quantitative and qualitative evaluation of MAS methodologies (Gaia, MaSE and Tropos). It explores the following aspects: concepts, properties, notations and modeling techniques, process and pragmatics. Dam and Winikoff (2004, 2013), (Dam, 2003) illustrate the strengths and weaknesses of MaSE, Prometheus and Tropos methodologies through an attribute-based evaluation process.

The Guardian Angel (GA) Exemplar proposed by Yu and Cysneiros (Yu and Cysneiros, 2002) defines a set of questions to evaluate the behaviour of MAS methodologies and is expressed in terms of a set of numbered scenarios. The GA is an easily comprehended open system that provides automated support to assess patients with chronic diseases through a set of "guardian angel" software agents.

The GA exemplar is a complete solution, with a practical, real and significant enough example, to test and verify how the methodology behaves in close-to-real situations. The primary concern of the exemplar is to highlight the strengths, weaknesses and potentials of each methodology justified by the artefacts (work products) that can be used to answer the methodology questions.

We chose to use the GA exemplar as it was the only one we found in the literature that proposes complex situations that can be used empirically to evaluate different methodologies that go beyond toy problems.

4 DOMAIN THEORY: AGENT-ORIENTED METHODOLOGIES

In order to define a Domain Theory for Agent-Oriented Methodologies, we have compiled the knowledge gathered from papers on AOSE methodologies listed in section 2 (Dam and Winikoff, 2013), (Luck, Mcburney and Preist, 2003), (Sturm and Shehory, 2014), (Tran and Low, 2005), (Elamy and Far, 2008) together with the results from our experience using the Guardian Angel exemplar over the past 10 years.

While building the MAS Ontology, we tried to answer the following research questions: (i) in what situations is a methodology or method fragment best applied?; (ii) which instruments are used to define the methodological questions from GA and from the works of (Dam and Winikoff, 2013), (Luck, McBurney and Preist, 2003), (Sturm and Shehory, 2014), (Tran and Low, 2005), (Elamy and Far, 2008) (iii) what are the general concepts of agents that a MAS methodology should support?; (iv) what are the specific concepts of agents that a MAS methodology can support?; (v) what are the notations and modeling techniques found in the methodology?; (vi) what are the support resources offered by the methodology?

4.1 Approach

In the ontology, we assembled the knowledge generated by using the GA exemplar pertinent to four different methodologies (Gaia, MaSE, Prometheus and Tropos). We organized the knowledge and experiences gained by signaling which work product is responsible for a certain task when answering the questions listed above while applying the exemplar to each of the aforementioned methodologies.

4.2 MethodBase GA: Experience Modeling with GA

The MethodBase GA is the knowledge base that compiles the work done over many years by MSc and last year undergrad Computer Science students. During this time, these students modeled multiagent systems using methodologies such as Gaia, MaSE, Prometheus and Tropos and relying on scenarios proposed in Guardian Angel exemplar. After modeling the solutions, the students answered the methodological issues in accordance with strengths, neutral or weakness, as seen in figure 1.

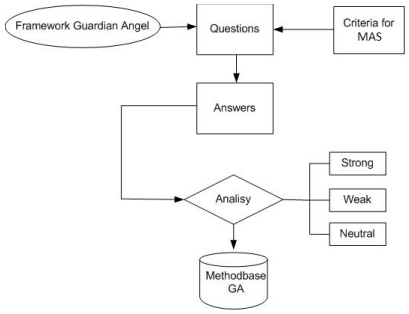


Figure 1: Methodbase GA.

4.3 Evaluated MethodBase: Evaluating Methodologies

The Evaluated MethodBase is the knowledge base built based on the work of Sturm and Dam (Sturm and Shehory, 2014), (Dam and Winikoff, 2013), as illustrated in figure 2. The proposed Evaluated MethodBase includes the following concepts: General Concepts of MAS, Specific Concepts of MAS, Notations, Modeling Techniques, Process and Pragmatics (practical aspects).

The ranking of values ranges from 0 to 6, where 0 represents cases where a certain characteristic is not applicable, 1 Refers to but not detailed, 2 Limited, 3 Neutral, 4 Small issues, 5 Minor deficiencies and 6 is the ideal efficiency. This was an adaptation of (Sturm and Shehory, 2004), (Dam and Winikoff, 2002) using the databases Methodbased GA and Evaluated Methodbase.

5 ONTOLOGY LEARNING FROM EXPERIENCE

Many definitions of ontology can be found in the literature. However, Sure (Sure, Staab and Studer, 2002) provides a simple and comprehensive definition: "An ontology is a formal and explicit specification of a shared conceptualization". In this definition "formal" means readable by computers; "explicit specification" refers to concepts, properties, relations, functions, constraints, axioms, explicitly defined; "shared" means consensual knowledge, and "conceptualization" refers to an abstract model of

some phenomenon in the world real. The ontology built in this work was based on a middle-out strategy (Uschold and King, 1995), in which concepts were generalized and specialized.

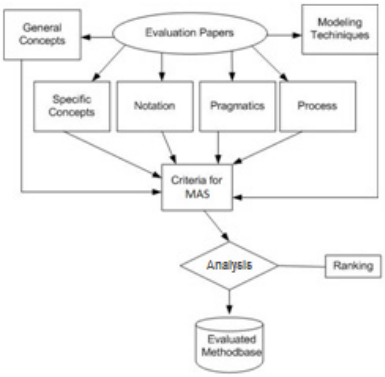


Figure 2: Evaluated Methodbase.

Building on identify concepts (terms) that can provide short assertive sentences, we developed an ontology based on the knowledge acquired from both databases Methodbased GA and Evaluated Methodbase.

The ontology development is defined through seven stages: *Ontology Specification, Knowledge Acquisition, Conceptualization, Formalization, Integration, Implementation* and *Evaluation*.

The *Ontology Specification* is used to prepare a document using natural language, containing information such as the primary ontology goal and its other purposes.

Knowledge Acquisition focuses on possible sources of knowledge. In this survey the GA experiences were used in order to manage the data collected, analyzed and categorized according to their degree of strength, weakness or neutrality.

Conceptualization focuses on structuring the domain knowledge into a conceptual model and was based on the acquired vocabulary in the previous phases, in order to describe the problems and their possible solutions

In the *Formalization*, the concepts are now formally written through OWL. The Protégé tool version 4.3 (Protege, 2000) was used, and the first preliminary version of the ontology was generated. At this stage, a taxonomy that shows the processes of a multiagent system is available.

The *Integration* stage obtains the representative experimental ontology from the Guardian Angel exemplar and is re-evaluated to better address the domain of multiagent systems.

At this stage, other studies on the comparison of methodologies are integrated. (Sturm and Shehory, 2004).

The *Implementation* used the Pellet, a Protégé plugin to automatically check the ontology consistency and also takes into account the experience of validating the data, as well as establishing the comparable relationship between the Methodbase GA and Evaluated Methodbase values, classes and attributes. Each phase of the ontology is related to models, tables or charts, which serve to guide the building process of the MAS Ontology, here defined as products, as seen in figure 3.

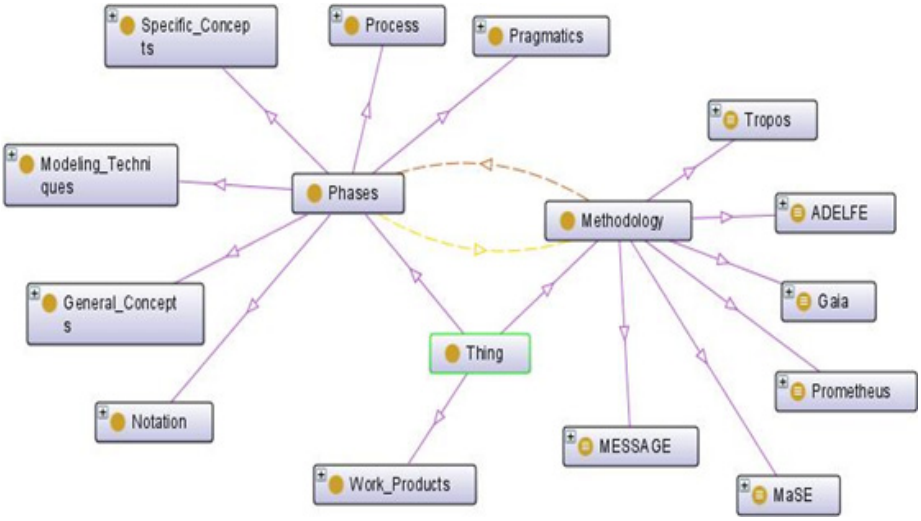


Figure 3: MAS Ontology.

6 MAS ONTOLOGY AND RESULTS

The MAS ontology has three main classes: Methodology, Phases and Work_Products.

The class Methodology focuses on the methodologies that are the study objects (Tropos, ADELFE, Gaia, Prometheus, MaSE, and MESSAGE).

The Phases class combines the characteristics essential to multiagent systems (General Concepts, Modeling Techniques, Notation, Pragmatics, Process and Specific Concepts). Each class has a set of attributes associated with it. (e.g. General Concepts attributes such as: Autonomy, Reactiveness, Sociality, Proactiveness, Reasoning, Mobility).

The class Work_Products lists the necessary artifacts to build a multiagent system.

Figure 3 shows a simplified MAS ontology. The Phase class is associated with the Methodology class. In this relationship, subclasses of Phase are related to subclasses of Methodology. The class Work_Products is also listed to illustrate the artefacts in Figure 4. It is important to determine which attributes from the Phases class might be associated with corresponding work products. For example, in Figure 5 the subclass Tropos_Products has two phases: Tropos_Analysis and Tropos_Design. Each subclass has its own subclasses. Tropos analysis is composed of Actors Diagram and Reasoning Diagram. Tropos Design consist of Extended Actors Diagram, Table of Actors and Capabilities, Table of Agents, Agents Interaction Diagram, Tasks or Plans Diagram and Capabilities Diagram.

6.1 Schematic Model

In order to facilitate understanding the domain of multiagent systems by ontological representation, a Schematic model of MAS Ontology (Figure 6) was developed to illustrate the relationship between classes, attributes and expected values.

In Figure 6 the schematic forms are described as follows:

- Ellipse Form - Classes or Subclasses
- Rectangle Form - Attributes
- Dotted Ellipse - Value types for attributes
- Dotted Rectangle Form - Class properties

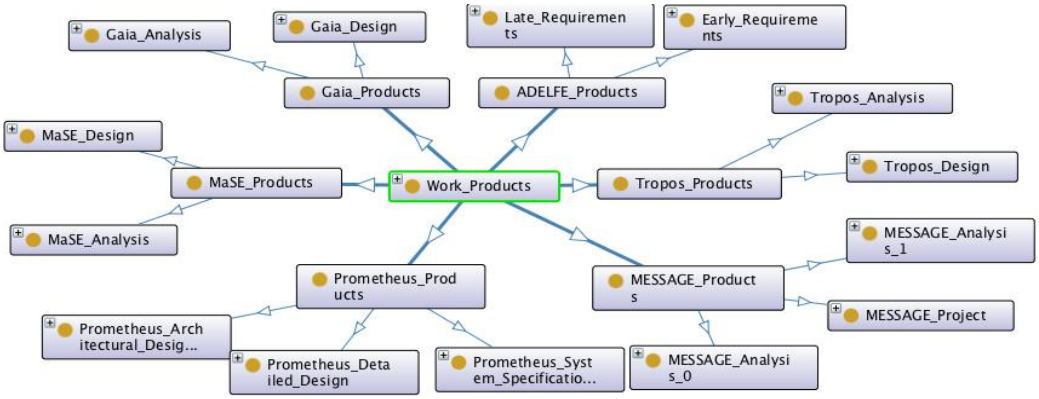


Figure 4: Work Products Detailed.

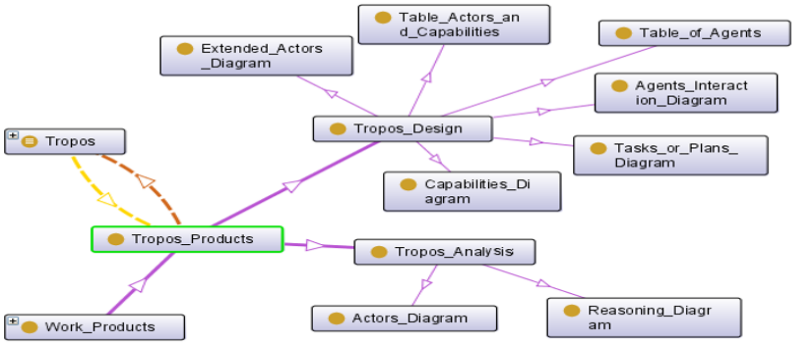


Figure 5: Tropos Work Products.

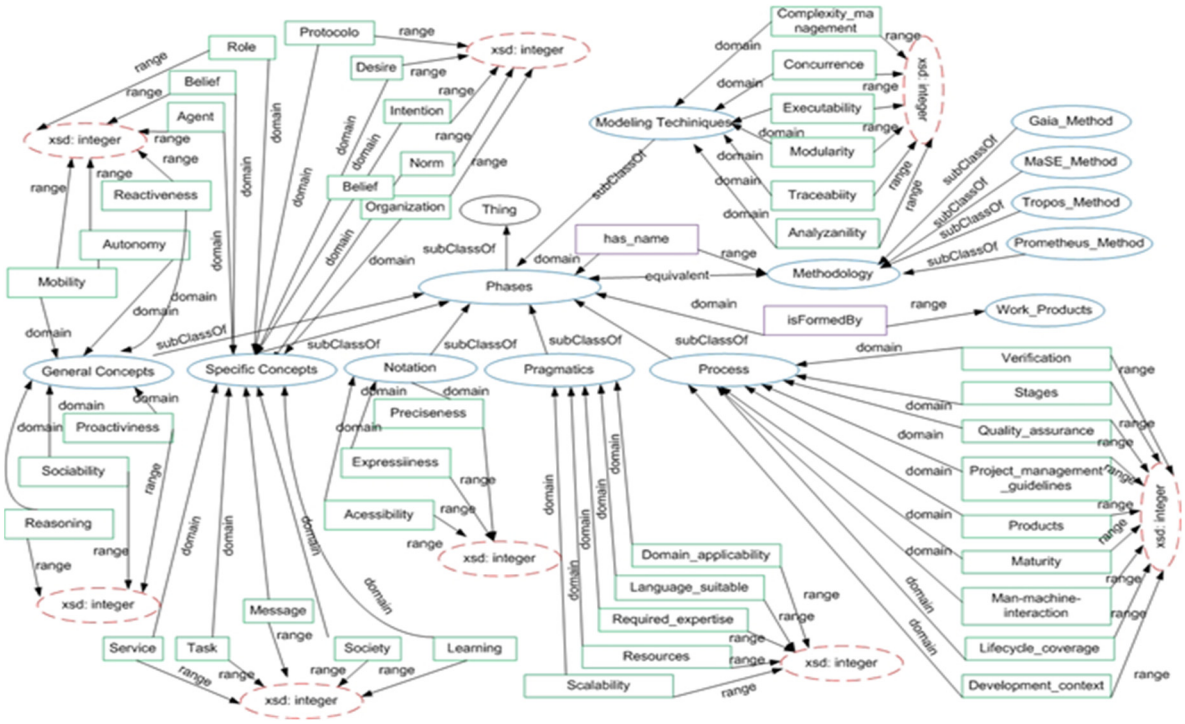


Figure 6: Schematic model of MAS Ontology.

6.2 Examples

For MAS Ontology population the individuals were separated into two groups: (i) representing specific values of GA attributes (Yu and Cysneiros, 2002) and (ii) representing the set of attributes that make up a methodology in the evaluation comparison papers (Sturm and Shehory, 2004, 2014), (Dam and Winikoff, 2002, 2013). Thereby the query may return a particular specific situation or a methodology.

Figure 7a represents an unsuccessful search carried out in plugin DL Query (Protege, 2000), where the attribute Mobility (The quality or state of being mobile) was defined as value 4, and no individual was found. Figure 8b represents a successful search done in plugin DL Query, where the attribute Mobility was defined as value 3

In this scenario, three methodologies were found (figure 7b). Figure 8 shows the associated work products (e.g. Mobility Tropos and Mobility Prometheus) obtained from a refined search for Tropos and Prometheus.

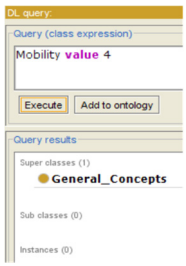


Figure 7.a: Unsuccessful Search.

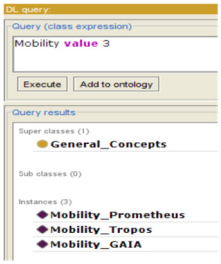


Figure 7.b: Successful Search.

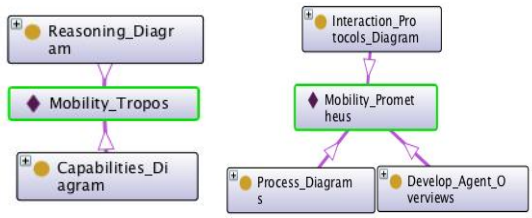


Figure 8: Associated Work Products.

7 CORRELATED WORKS

Several works addressing the evaluation of agent orientation methodologies have been published (Dam and Winikoff, 2002), (Sturm and Shehory, 2004), (Dam, 2003), (Dam and Winikoff, 2004), (Tran and Low, 2005), (Elamy and Far, 2008), (Iglesias, Garijo and González, 1999), (Cernuzzi, Rossi and Plata, 2002), (Sure, Staab and Studer, 2002), (Casare et al, 2014). They consist of quantitative and qualitative evaluation framework based on checklists of certain properties, qualities, attributes or characteristics of the methodology and some simple problems.

Tran and Low (2005, 2005a) compared ten methodologies (Gaia, Tropos MAS CommonKADS, Prometheus, Passi, ADELFE, MaSE, RAP, MESSAGE, Ingenius). They used a criteria checklist that was developed to assess the resources of the chosen methodologies, covering the process, techniques and model stages.

Cernuzzi and Zambonelli (2011) used the multivalued statistical method for quantitative evaluation of profiles. The goal was to present the potential profile analysis in the comparison process for the evaluation of methodologies, searching for similar evaluations to confirm the results.

Our study differs from similar works by proposing the use of a knowledge base where the knowledge is expressed and organized as an ontology. The ontology can guide the developer to select fragments of methodologies that best fit the multiagent system under development. It allows for queries to be made that can help developers to customize their development process. It helps them to search for where the methodologies best fit their needs considering the specific project at hand.

On another level, it also helps researchers further developing these methodologies to easily compare where their approaches fall behind when compared to other existing methodologies and therefore, where they should invest more effort to develop furthere their methodologies.

8 CONCLUSIONS

As a result of the fast dissemination of MAS methodologies, deciding what methodology to use in a project is a complex task. Many frameworks and toolkits are provided, but they do not always offer support to assist developers in choosing the best or most appropriate methodology to handle the project at hand. This paper proposes an ontology-based support to help developers faced with the need to use agent-oriented properties to develop software. The ontology was created based on the experience gathered by applying the Guardian Angel exemplar in four agent-oriented software engineering methodologies, as well as adding the knowledge obtained from the results from Sturn and Dam (2004) and Dam (2003). The knowledge base provided in this ontology can assist developers to use these methodologies and also to choose better the adequate artifacts for a particular domain.

The MAS Ontology approach focuses on being a facilitator for developing a MAS process, as it concentrates on relationships between the principles of software engineering evaluation and experience. Furthermore, it can be extended to suit the particularities of other AOSE methodologies and other studies based on statistics, as in (Iglesias, Garijo and González, 1999).

Future works will address a systematic validation of the Ontology using case studies where different groups of randomly selected students will be asked to develop solutions to a specific problem. Some students will use the ontology, and another set of students will have to develop the solution using pre-determined methodology. Final results will then be compared.

REFERENCES

Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., & Mylopoulos, J. (2004). Tropos: An agent-oriented software development methodology. *Autonomous Agents and Multi-Agent Systems*, 8(3), 203-236.

Casare, S. J., Brandão, A. A., Guessoum, Z., & Sichman, J. S. (2014). Method Framework: a situational approach for organization-centered. *MAS. Autonomous agents and multi*-agent systems, 28(3), 430-473.

Cernuzzi, L., & Zambonelli, F. (2011). Improving comparative analysis for the evaluation of AOSE methodologies. International *Journal of Agent-Oriented Software Engineering*, 4(4), 331-352.

Cernuzzi, L., Cossentino, M., & Zambonelli, F. (2005). Process models for agent-based development.

Engineering Applications of Artificial Intelligence, 18(2), 205-222.

Cernuzzi, L., Rossi, G., & Plata, L. (2002, November). On the evaluation of agent oriented modeling methods. In *Proceedings of Agent Oriented Methodology Workshop*, Seattle, (Vol. 29).

Coburn, M. (2000). *Jack intelligent agents: User guide version 2.0*. AOS Pty Ltd..

Cossentino, M., Fortino, G., Garro, A., Mascillaro, S., & Russo, W. (2008). PASSIM: a simulation-based process for the development of multi-agent systems. *International Journal of Agent-Oriented Software Engineering*, 2(2), 132-170.

Cossentino, M., Gaglio, S., Galland, S., Gaud, N., Hilaire, V., Koukam, A., & Seidita, V. (2009). A MAS metamodel-driven approach to process fragments selection. In *Agent-Oriented Software Engineering IX* (pp. 86-100). Springer Berlin Heidelberg.

Dam, K. H. (2003). *Evaluating and comparing agent-oriented software engineering methodologies* (Doctoral dissertation, School of Computer Science and Information Technology, RMIT University, Australia).

Dam H. K., Winikoff M., 2013. Towards a next-generation AOSE methodology. *Science of Computer Programming*, v. 78, n. 6, p. 684-694.

Dam, K. H., & Winikoff, M. (2004, January). Comparing agent-oriented methodologies. In *Agent-Oriented Information Systems* (pp. 78-93). Springer Berlin Heidelberg.

Deloach, S. (2004). The MaSE methodology. *Methodologies and software engineering for agent systems*, 107-125..

DeLoach, S. A. (2001). *Analysis and Design using MaSE and agentTool*. Air force inst of tech wright-patterson afb oh school of engineering and management.

Elamy, A. H. H., & Far, B. (2008). On the evaluation of agent-oriented software engineering methodologies: a statistical approach. In *Agent-Oriented Information Systems IV* (pp. 105-122). Springer Berlin Heidelberg.

Iglesias, C. A., Garijo, M., & González, J. C. (1999). A survey of agent-oriented methodologies. In *Intelligent Agents V: Agents Theories, Architectures, and Languages* (pp. 317-330). Springer Berlin Heidelberg.

Luck, M., McBurney, P., & Preist, C. (2003). *Agent technology: enabling next generation computing (a roadmap for agent based computing)*. AgentLink/University of Southampton.

Munroe S. et al., 2006. Crossing the agent technology chasm: Lessons, experiences and challenges in commercial applications of agents. *The Knowledge Engineering Review*, v. 21, n. 04, p. 345-392.

OMG Group. (2008). *Software & Systems Process Engineering Meta-Model Specification*, at http://www.omg.org/spec/SPEM/2.0/

Padgham, L., & Winikoff, M. (2002, November). Prometheus: A pragmatic methodology for engineering intelligent agents. In *Proceedings of the OOPSLA 2002 Workshop on Agent-Oriented Methodologies* (pp. 97-108).

Padgham, L., & Winikoff, M. (2003). Prometheus: A methodology for developing intelligent agents. In *Agent-oriented software engineering III* (pp. 174-185). Springer Berlin Heidelberg.

Pěchouček M., Mařík V., 2008. Industrial deployment of multiagent technologies: review and selected case studies. *Autonomous Agents and Multiagent Systems*, v. 17, n. 3, p. 397-431.

PROTEGE (2000). The Protege Project. at http://protege.stanford.edu.

Sturm, A., & Shehory, O. (2004, January). A framework for evaluating agent-oriented methodologies. In *Agent-Oriented Information Systems* (pp. 94-109). Springer Berlin Heidelberg.

Sturm, A., & Shehory, O. (2014, January). The landscape of agent-oriented methodologies. In *Agent-Oriented Software Engineering* (pp. 137-154). Springer Berlin Heidelberg.

Sure, Y., Staab, S., & Studer, R. (2002). Methodology for development and employment of ontology based knowledge management applications. *ACM SIGMOD Record*, 31(4), 18-23.

Tran, Q. N. N., & Low, G. C. (2005). Comparison of ten agent-oriented methodologies. *Agent-oriented methodologies*, 341-367.

Tran, Q. N. N., Low, G., & Williams, M. A. (2005). A preliminary comparative feature analysis of multi-agent systems development methodologies. In *Agent-Oriented Information Systems* II (pp. 157-168). Springer Berlin Heidelberg.

Uschold, M., & King, M. (1995). *Towards a methodology for building ontologies* (pp. 15-30). Edinburgh: Artificial Intelligence Applications Institute, University of Edinburgh.

Wooldridge, M., Jennings, N. R., & Kinny, D. (2000). The Gaia methodology for agent-oriented analysis and design. *Autonomous Agents and multi-agent systems*, 3(3), 285-312.

Yu, Eric (2009). Social Modeling and i*. In *Conceptual Modeling: Foundations and Applications* (pp. 99-121). Springer Berlin Heidelberg.

Yu, E., & Cysneiros, L. M. (2002, May). *Agent-oriented methodologies-towards a challenge exemplar*. In Proc of the 4 Intl. Bi-Conference Workshop on AOIS, Toronto (Vol. 151).

Zambonelli, F., Jennings, N. R., & Wooldridge, M. (2003). Developing multiagent systems: The Gaia methodology. *ACM Transactions on Software Engineering and Methodology* (TOSEM), 12(3), 317-370.

Requirements Engineering and Variability Management in DSPLs Domain Engineering: A Systematic Literature Review

Léuson M. P. da Silva[1], Carla I. M. Bezerra[1,2,3], Rossana M. C. Andrade[2,3] and José Maria S. Monteiro[2]

[1]*Federal University of Ceará (UFC), Quixadá Campus, ZIP: 63902-580, Quixadá, CE, Brazil*

[2]*Federal University of Ceará (UFC), Pici Campus, ZIP: 60455-760, Fortaleza, CE, Brazil*

[3]*Group of Computer Networks, Software Engineering and Systems (GREat), Pici Campus, Bloco 942-A, ZIP: 60455-760, Fortaleza, CE, Brazil*

leuson@alu.ufc.br, {carlabezerra, rossana}@great.ufc.br, monteiro@lia.ufc.br

Keywords: Dynamic Software Product Line, Requirements Engineering, Variability Management.

Abstract: Recently, Software Product Lines (SPLs) have been used successfully for building products families. However, the currently and complex software products demand more adaptive features. Today, many application domains demand capabilities for flexible adaptation and post-deployment reconfiguration. In this context, Dynamic Software Product Lines (DSPLs) represent a way to produce software products able to change their own behavior at runtime due to the changes in the product use environment. DSPLs present some interesting properties such dynamic variability and reconfiguration at runtime. The dynamic variability is represented by the definition of variants and context information. The reconfiguration at runtime is the process that enables the features activation and deactivation in a configuration product. Both properties are closely related to the requirements engineering and variability management, in the domain engineering life-cycle. In this research, we provide a systematic literature review that aims to identify the activities, assets, tools and approaches that are used in requirements engineering and variability management in DSPLs domain engineering. We performed a manual and automatic search, resulting in 581 papers of which 37 were selected. We also provide a discussion about the challenges and solutions of runtime variability mechanisms in the context of DSPLs.

1 INTRODUCTION

Software Product Lines (SPLs) can be defined as a set of software-intensive systems sharing a common and managed set of features that satisfies the specific needs of a particular market segment or mission (Northrop et al., 2007). However, SPLs just support the development of static products (Hinchey et al., 2012), i.e., SPLs products are not able to adapt their own behavior to the changes in the users needs at runtime (Bencomo et al., 2012). On the other hand, the currently complex systems need to deal with dynamic aspects, such as successive reconfigurations at runtime, after their first deployment (Bosch et al., 2015). In this context, emerged Dynamic Software Product Lines (DSPL) (Hallsteinsen et al., 2008).

DSPLs extend existing product line engineering approaches by moving their capabilities to runtime (Hinchey et al., 2012). In DSPLs, products can be reconfigured dynamically at runtime after their initial derivation (Bencomo et al., 2012). Although DSPLs have some differences compared with SPLs, DSPLs still share the same development life-cycle as presented by Capilla et al. (Capilla et al., 2014a).

DSPLs, as well SPLs, are composed of two main development life-cycle: domain and application engineering (Hallsteinsen et al., 2008). The domain engineering is responsible for (i) specifying, documenting and developing the assets that will be used to compose the future products of the line. Besides it is responsible for (ii) producing the necessary SPLs infrastructure, composed of: a common architecture and its variation points, a set of reusable parts and a model to represent the variability (Bencomo et al., 2012).

Once DSPLs adapt their own behavior at runtime, besides to identify the requirements, it is necessary to recognize the contexts that the line will need to support. These tasks are performed in the domain engineering life-cycle, through two different activities: domain and context analysis (Capilla et al., 2014a). The domain analysis specifies the domain that the line will support, identifying and documenting the vari-

Silva, L., Bezerra, C., Andrade, R. and Monteiro, J.
Requirements Engineering and Variability Management in DSPLs Domain Engineering: A Systematic Literature Review.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 544-551
ISBN: 978-989-758-187-8

able features of the domain (Capilla et al., 2014b). The context analysis captures the contexts to be supported by the DSPL (Capilla et al., 2014b). An important activity from context analysis is to identify the information used by the products reconfiguration process that happens due to the changes in use environment. When a new context is identified, the product needs to check which features must be activated and which should not (Capilla et al., 2014a).

Guedes et al. (Guedes et al., 2015) present a systematic mapping focused on DSPLs aspects to identify methodologies that are used to execute the variability management. However, the results do not present what activities are used to project and how these activities need to be executed to ensure the DSPLs variability was well understood. In the work of Da Silva et al. (da Silva et al., 2013) is presented a SLR that aims at understanding how dynamic derivation is made in DSPL. The work identifies how the models, approaches and methods are used to address the dynamic derivation problem in DSPLs, but it is not presented how these assets are made, what information, roles and activities are involved to define them.

In this context, we performed a Systematic Literature Review (SLR) to investigate how the requirements engineering and variability management are performed in DSPLs domain engineering. We aimed to identify the activities, assets, tools and approaches that are used. As result, we analyzed 37 studies dated from 2008 to 2015. The main contribution of this work is a catalogue of activities to support the requirements process in DSPLs domain engineering.

The remainder of this work is organized as follows. Section 2 reports the SLR. Section 3 describes the studies classification. The results of each research questions are presented in section 4. Section 5 presents a discussion about the results. Section 6 discusses the threats of validity. Section 7 concludes this work and presents suggestions for future work.

2 SYSTEMATIC REVIEW PROCESS

The review process of this work followed the guidelines of (Keele, 2007) and (Kitchenham et al., 2009). The process included the definition of three activities: planning, conducting and results reporting. In the planning was defined the review protocol and in the conducting, the focus was on the selection and analyses process of the work. Finally, the results reporting comprised the results presentation.

To support the SLRs execution we use a tool to automatize some steps. The adopted tool was StArt [1] (State of the Art through Systematic Reviews) that supports the review process since the definition of the review protocol until the results report. Due to StArt be a desktop tool, some steps were supported by templates allowing the parallel work among the researchers. The next subsections present how the review process was done.

2.1 Research Questions

This work followed a main research question and six (6) secondary questions. The first four secondary questions are related to list the activities, assets, tools and the approaches used to represent the variability and requirements in DSPLs domain engineering. Additionally, the last two research questions deal with the DSPLs variability and aim to specify what information are used, still in the requirements engineering, to do or just to support, the variability management at runtime.

- RQ1. How are the requirements engineering (RE) and variability management (VM) executed in DSPLs domain engineering?
  - RQ1.1. What activities of RE and VM are used in DSPLs?
  - RQ1.2. What approaches are used to document the requirements in DSPLs?
  - RQ1.3. What assets are built of RE and VM in DSPLs?
  - RQ1.4. What approaches are used to represent the DSPLs variability?
  - RQ1.5. What approaches are used to support the reconfiguration process in DSPLs?
  - RQ1.6. What approaches are used to eliminate possible inconsistencies in DSPLs variability model?

2.2 Search Process

The search process started with a manual and automated search, in digital libraries (DL). The adopted DLs were: Scopus [2], Compendex[3] and Web of Science[4]. The manual search was necessary due to the verification that the automated search did not return papers from important conferences dated from 2015 (SPLC, VaMoS). After the two searches, all identified papers were joined at the same work set.

[1] http://lapes.dc.ufscar.br/tools/
[2] http://www.scopus.com/
[3] http://www.engineeringvillage.com/
[4] https://apps.webofknowledge.com/

To perform the automated search, we used a search string. This string was defined through keywords, extracted from each research question. At the end, the selected words were joined with search operators (AND, OR). To ensure the string was appropriated to return valid papers, we tested it many times. The adopted search string is presented bellow:

("Software product line engineering" OR "Domain Engineering" OR "Domain Analysis" OR "Context Analysis") AND (Requirements OR "Requirements Engineering" OR Elicitation OR Analysis OR Specification OR Verification OR Management) AND ("feature model" OR "variability model" OR "decision model" OR "domain model") AND ("Software Product Line" OR "product family") AND (autonomic OR pervasive OR ecosystems OR dynamic OR "context-aware" OR adaptive)*

2.3 Inclusion/Exclusion and Quality Criteria

The papers of the automated and manual search were analyzed according to some criteria. To justify the reason that a paper would be selected or not, we determine inclusion and exclusion criteria. Additionally, the papers content should be evaluated. It was done following the guidelines of Kitchenham et al. (Kitchenham et al., 2006). For that, we define 6 questions and a valuation function. The evaluation score could vary from 0, for papers that do not satisfy a criterion, to 6, for papers that satisfy totally a criterion. The inclusion/exclusion criteria are listed in Table 1.

Table 1: Inclusion/Exclusion criteria.

Inc/Exc	Criterion
Inc.	Activities about RE and VM for DSPLs domain engineering
Inc.	Approaches for the RE and VM activities
Inc.	Approaches to support the requirements and variability
Inc.	Approaches to support the inconsistencies treatment in DSPLs variability model
Exc.	Do not focus on RE and VM of DSPLs
Exc.	Not written in English
Exc.	Out of the valid formats (papers from conferences and journals)
Exc.	Could not be accessed from UFC's network or by contacting authors

2.4 Data Extraction

The adopted approach for data extraction was based on the work of Montagud et al. (Montagud et al., 2012). For each research question was defined some values that could be the answers presented by the papers. It was done to make easy the extraction process. The information to be extracted were: activities from domain and context analysis, extracted from Capilla et al. (Capilla et al., 2014b), involved roles, built assets, variability representation, inconsistency treatment, validation type and related adopted method, and context use.

3 STUDIES CLASSIFICATION

The studies classification process corresponds to the conducting and review reporting phases of the SLR. This process was done supported by four researchers and made in 5 steps:

- Step 1: Perform string search on DLs and the manual search;
- Step 2: Elimination of duplicated papers using the StArt tool;
- Step 3: Title, Abstract and Keyword analysis of all papers according to the inclusion/exclusion criteria;
- Step 4: Full reading and analysis of the papers according to the inclusion/exclusion criteria by the researchers through pairs read;
- Step 5: Last check for duplicated papers identification, and information extraction and quality evaluation execution.

If during the classification process of a paper a nonconformity among the researchers happens, the researchers are responsible for solving the conflict and decide together if the paper would be eliminated or not. An overview of this process is presented in Figure 1. It is possible to see the number of returned papers of each source and the number of selected papers after each step, respectively. Finally, after the step five, the set of papers decreased to 37 papers that were analysed again to extract the principal information.

4 RESULTS

This section presents the SLR results from the 37 selected work. The papers of our work were selected in August/2015. Due to it some papers from important

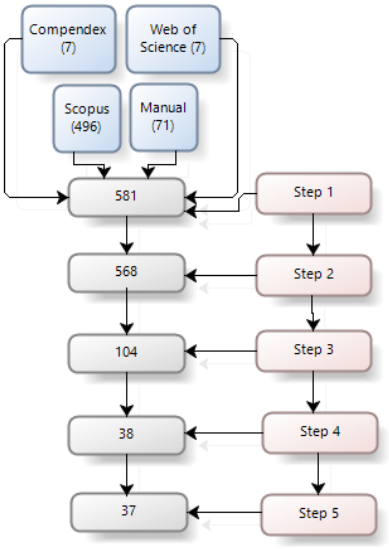


Figure 1: Study selection process.

sources dated after august are not presented here. The next subsections present each adopted research question and their related results.

4.1 Papers Overview

Before the analysis of each research question is important to present an overview about the selected papers. Figure 2 presents the number of selected papers along the years according to the three selection process. According to Figure 2 is possible to see that the oldest selected work in the 3rd selection dates from 2008. This means that the researches in DSPLs is still a new research area. From 2011 the number of selected papers per year is more regular varying between 4 and 7 papers.

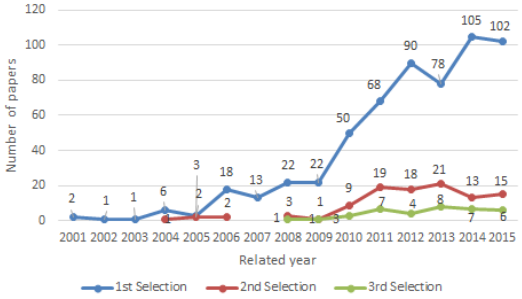


Figure 2: Selected papers along the years.

About the papers use context, the majority (29) was developed to academic ends, while 8 papers were focused on the industrial area. It reinforces the idea of DSPL is a new research area, but there are some work that promote its use in industry. About the quality evaluation of the papers only one has the maximal score 6 (S07), while the majority has a score between 3 and 5. Table 6 presents the the quality evaluation of each paper.

4.2 Requirements Engineering and Variability Management in DSPLs Domain Engineering (RQ1)

The main research question aims to know how the requirements engineering and variability management are executed in DSPLs domain engineering. The results of each secondary question are presented as follows.

4.2.1 RQ1.1 What Activities of RE and VM are used in DSPLs?

Table 2 shows the identified activities. As can be seen, the activities are separated in three groups (phases). The first group brings the activities commonly used in the traditional software development, domain analysis. The second group presents the activities of the context analysis. Some papers present the activity Define operational rules of the domain analysis phase that is related to the architecture modelling. This kind of activity is commonly executed on the project domain phase, like some papers execute. This activity is attributed to the the second group, when it is executed on the domain analysis phase, and attributed to the third group, when a paper performs this activity on the project domain phase. A finding is that the activity Define multiple connection is executed only in the project domain phase.

Although the papers present specific activities to attend to the domain specification, some details about these activities execution were clearer in the section of studies cases. However, the identified activities are more related to requirements and variability modeling, and how the variability management is done at design time and runtime. Activities about requirements elicitation are executed with fewer importance.

This question identified too the roles involved in the activities, shown in Table 2. Just 9 papers present the roles involved in their activities. Although the roles are more involved in the project specification and modelling, there is not a clear definition about the responsibilities of each one. For example, the designers and the architects are responsible for modeling the DSPLs architecture, S05 and S09. The analysts are involved in the activities of modeling such as identifying of features, S06 e S07. On the other hand, the definition of the developer role just combines the responsibilities of the other roles. It difficulties the understanding about which roles are necessary to ex-

Table 2: Result from the first research question.

Domain Engineering			
	Domain Analysis		Domain Project
	Domain Analysis	Context Analysis	Domain Project
Activities	Conception, Elicitation, Specification, Validation	Identify physical properties, Identify context features, Model context features, Define operational rules	Define operational rules, Define multiple connection
Roles	Analyst (S06, S07), Designer (S05, S06, S15, S16, S26, S35), Architect (S09), Developer (S31)		

ecute each activity.

The identified activities are supported by some tools. Most papers (64%) did not use tools to support their process or they did not mention it. The rest of the papers use tools to project modelling as also to verify the consistency and correctness of the models. Table 3 presents the tools related to each purpose.

Table 3: Tools support.

Purpose	Tools and Papers
Workflow representation	BPMN (S10)
Variability modelling	Fama (S02), MOSkitt4SPL (S07), Atlas Model (S07), BPMN (S10), Familiar (S26), FeatureIDE (S26), eMoflon (S8), Odyssey (S28), DOPLER (S33), VariaMos (S36, S37)
Verification models	GNU Prolog (S07), Clafer (S10), Familiar (S26), VariaMos (S36, S37).

4.2.2 RQ1.2 What Approaches are used to Document the Requirements in DSPLs?

Only 6 papers present approaches that are used to document the requirements. The adopted approaches vary of the use UML diagrams, class diagram and use case (S13), sequence diagram (S15), approach that does not support the concept of variability, until the use of new approaches like Schemas (S25) and Goals (S35). S01 and S05 organize the domain requirements in feature model, but they do not present what approaches are used to transform the requirements, textual specification, in features of the variability model.

4.2.3 RQ1.3 What Assets are Built of RE and VM in DSPLs?

Table 4 presents all identified assets. The feature model as also the context feature model represent the assets with the higher frequency, 37% each one. The feature model is used to represent the DSPLs variability, as is used by traditional SPLs. The deference in DSPLs is about the context feature model that joins the context features, necessary to the reconfiguration process. The next more used model is the extended feature model (6 papers), this model extends the feature model to support specific needs.

Table 4: Built Assets.

Assets	Papers
Aspect model	S23, S31
Base composition model and Composition model	S07
Base Model	S10
Context feature model	S01, S06, S07, S10, S11, S12, S20, S22, S28, S29, S30, S32, S34, S37
Extended feature model	S05, S08, S10, S16, S18, S34
Feature model	S01, S02, S03, S04, S07, S12, S13, S20, S21, S26, S27, S28, S32, S37
Feature model adaptation	S01
Requirements specification	S13, S15, S25, S35
Rules adaptions	S24
States machine	S08
Weaving model	S07, S10

4.2.4 RQ1.4 What Approaches are used to Represent the DSPLs Variability?

The models used to represent the variability were: Aspect model (S31), Actor model (S17), MVRP (S16), OWL (S12), OCL (S04) and Feature Model (S01, S02, S03, S07, S10, S11, S13, S14, S15, S19, S21, S23, S26, S27, S28, S29, S34, S37).

The feature model represents the most used approach to represent the variability due to its use flexibility. This approach allows to change its own properties in order to attend the new use needs. Other consideration about the feature model is that there is not a pattern to represent the variability. For example, some papers identify context features but the way to organize these information vary. Some put the context features at the same feature model used to represent the variability while others put it in a independent model.

4.2.5 RQ1.5 What Approaches are used to Support the Reconfiguration Process in DSPLs?

This question is related to the approaches that are used to support the adaptability at runtime. This activity is more related to the domain design. Table 5 presents the results of this question.

Table 5: Approaches to reconfiguration process.

Approaches	Studies
Adaptation rules	S07, S24
ECA	S01, S02, S22, S27, S31, S32
MAPE-K	S15, S36
MAPE	S01, S35
Transformation rules	S26
Aspect models	S23
Context mapping	S10
PrtNets	S30
Contraint-satisfaction	S21

ECA (Event-Condition-Action) approach is the most widely used approach. ECA is based on rules that are created from constraints of different sources, such as the activity responsible for identifying operational rules (see subsection 4.2.1). The process of specifying the adaptation rules through MAPE (Monitor, Analyze, Plan and Execute) and MAPE-K (Monitor, Analyze, Plan, Execute and Knowledge) follows almost the same methodology for both. It is necessary to identify how the context information would be accessible, specifying the constraints that would be responsible to decide to what new context the product would change and what parts would be necessary to support this new context.

4.2.6 RQ1.6 What Approaches are used to Eliminate Possible Inconsistencies in DSPLs Variability Model?

In DSPLs it is necessary to ensure that not only the variability model at design time is consistent as also the consistency of the model when a reconfiguration happens. Although this consistency checking be an important step to ensure and validate the requirements, only 12 papers mentioned it. Eight papers (S07, S08, S10, S20, S22, S26, S28, S37) executed checks just to ensure the adopted variability model did not have inconsistencies. It was used tools to execute this checking (the list of identified tools is presented in Table 3).

Related to the papers that did checking at runtime (S02, S05, S06, S08, S15), only specific papers (e.g., S08, S15) present the approaches they adopted. The checking process followed the use of adaptation rules and the application of algorithms that are responsible for evaluating if a new configuration has any inconsistencies or not. The other papers just cite they did it but they did not present details about.

5 DISCUSSION

The domain engineering is responsible for exploring the domain that the DSPL will support. The results of this work show that DSPLs research has produced a set of assets, tools and approaches to support the activities necessary for DSPL requirements engineering and variability management.

About the activities of domain analysis, we concluded that the activities responsible for the domain specification and modeling have more importance that the others, like conception and elicitation. Because of it most papers do not specify formally the requirements domain. It goes against the domain analysis goal. The same problem happens with the activities of context analysis. The activities focus on the contexts identification and modeling, while the activities that treats the rules definition, responsible for identification of changes in a context, receives less importance. Still about domain analysis, we identified that the papers do not present activities to support the requirements changes as also do not present how the changes are treated, when they happen.

A challenge about the reconfiguration process is to change the variability model at runtime when a reconfiguration happens. Tools are used to check the model structure and its consistence but it was not identified yet a tool that supports this verification at runtime. Another important finding is about how the non-functional requirements (NFR) are identified and treated in domain engineering. Only one paper (S15) treated this issue with the same importance that functional requirements (FR) receive. The NFRs are more related to the domain project, that is when the architecture is modeled. Although this relation between the architecture modeling and the NFRs, it is important that the activity of architecture modeling receives the necessary information about the NFRs from domain analysis instead to identify the motivations around them. Other finding about NFRs is how the reconfiguration process is done. The NFRs are not considered by the papers in this process.

S09 is interested in evaluating the DSPLs quality attributes but they just focused on the features model. Evaluating the quality attributes in all DSPLs domain engineering using others assets represents a new research opportunity.

6 THREATS TO VALIDITY

This section discusses threats that could have affected the SLR results. We verified the threats were about the construct validity. Construct validity is concerned whether the treatments reflect the cause and the outcome reflects the effect (Wohlin et al., 2012).

The first possible threat is about the search string. It needs to reflect the main objective of the study otherwise would be returned work out of the study area. We have tried to minimize it through successive searches in DLs. As result, we verified that always some important papers, that we knew before, were returned (S06, S13). The second threat might have been about the search process. We verified papers from important conferences dated from 2015 were not been returned by the DLs. To solve this, we did a manual search in the proceedings of these conferences

7 CONCLUSIONS AND FUTURE WORK

This SLR aimed to determine how the requirements engineering and variability management supports the DSPLs domain engineering. To answer this question, we identified relevant and current studies following a formal approach. These studies were analyzed, evaluated and the information were extracted.

The results show the activities are concentrated on DSPLs modeling and specification. Traditional approaches (UML diagrams) can be used to document the domain requirements as also the feature model, that can be also used to represent the domain variability. The assets built in the activities can be done using tools responsible for modeling and consistence verification of them. We identified the following research opportunities: modeling and treatment of NFRs still in domain analysis; evaluating quality attributes using assets different of variability models; a mechanism to check the consistence and that be able to modify the variability model at runtime; defining a pattern to represent the variability in DSPLs; and exploring approaches that can be used for eliciting the FRs.

As future work, we would like to determine approaches to support the found gaps and to define a formal process for DSPLs RE and VM.

REFERENCES

Bencomo, N., Hallsteinsen, S., and Santana de Almeida, E. (2012). A view of the dynamic software product line landscape. *Computer*, 45(10):36–41.

Bosch, J., Capilla, R., and Hilliard, R. (2015). Trends in systems and software variability. *IEEE Software*, (3):44–51.

Capilla, R., Bosch, J., Trinidad, P., Ruiz-Cortés, A., and Hinchey, M. (2014a). An overview of dynamic software product line architectures and techniques: Observations from research and industry. *Journal of Systems and Software*, 91:3–23.

Capilla, R., Ortiz, O., and Hinchey, M. (2014b). Context variability for context-aware systems. *Computer*, (2):85–87.

da Silva, J. R. F., Pereira da Silva, F. A., do Nascimento, L. M., Martins, D. A., and Garcia, V. C. (2013). The dynamic aspects of product derivation in dspl: A systematic literature review. In *Information Reuse and Integration (IRI), 2013 IEEE 14th International Conference on*, pages 466–473. IEEE.

Guedes, G., Silva, C., Soares, M., and Castro, J. (2015). Variability management in dynamic software product lines: A systematic mapping. In *Components, Architectures and Reuse Software (SBCARS), 2015 IX Brazilian Symposium on*, pages 90–99. IEEE.

Hallsteinsen, Svein andHinchey, M., Park, S., and Schmid, K. (2008). Dynamic software product lines. *Computer*, 41(4):93–95.

Hinchey, M., Park, S., and Schmid, K. (2012). Building dynamic software product lines. *Computer*, (10):22–26.

Keele, S. (2007). Guidelines for performing systematic literature reviews in software engineering. In *Technical report, Ver. 2.3 EBSE Technical Report. EBSE*.

Kitchenham, B., Brereton, O. P., Budgen, D., Turner, M., Bailey, J., and Linkman, S. (2009). Systematic literature reviews in software engineering–a systematic literature review. *Information and software technology*, 51(1):7–15.

Kitchenham, B., Mendes, E., and Travassos, G. H. (2006). A systematic review of cross-vs. within-company cost estimation studies. In *Proceedings of the 10th international conference on Evaluation and Assessment in Software Engineering*, pages 81–90. British Computer Society.

Montagud, S., Abrahão, S., and Insfran, E. (2012). A systematic review of quality attributes and measures for software product lines. *Software Quality Journal*, 20(3-4):425–486.

Northrop, L., Clements, P., Bachmann, F., Bergey, J., Chastek, G., Cohen, S., Donohoe, P., Jones, L., Krut, R., Little, R., et al. (2007). A framework for software product line practice, version 5.0. *SEI.–2007–http://www. sei. cmu. edu/productlines/index. html*.

APPENDIX

Table 6: Quality evaluation of the studies.

ID	Reference	Quality Score
S01	L. Shen, X. Peng, and W. Zhao. Software product line engineering for developing self-adaptive systems: Towards the domain requirements. In Computer Software and Applications Conference (COMPSAC), 2012 IEEE 36th Annual.	4,25
S02	A. S. Nascimento, C. M. Rubira, and F. Castor. Arcmape: A software product line infrastructure to support fault-tolerant composite services. In High-Assurance Systems Engineering (HASE), 2014 IEEE 15th International Symposium.	3,25
S03	D. Dermeval, T. Tenorio, I. I. Bittencourt, A. Silva, S. Isotani, and M. Ribeiro. Ontology-based feature modeling: An empirical study in changing scenarios. Expert Systems with Applications, 42(11):4950–4964, 2015.	3,75
S04	C. Dubslaff, C. Baier, and S. Kluppelholz. Probabilistic model checking for feature-oriented systems. In Transactions on Aspect-Oriented Software Development XII, pages 180–220. Springer, 2015.	3,5
S05	R. Mizouni, M. A. Matar, Z. Al Mahmoud, S. Alzahmi, and A. Salah. A framework for context-aware self-adaptive mobile applications spl. Expert Systems with applications, 41(16):7549–7564, 2014.	5,25
S06	R. Capilla, J. Bosch, P. Trinidad, A. Ruiz-Cortes, and M. Hinchey. An overview of dynamic software product line architectures and techniques: Observations from research and industry. Journal of Systems and Software, 91:3–23, 2014.	4,5
S07	G. H. Alferez, V. Pelechano, R. Mazo, C. Salinesi, and D. Diaz. Dynamic adaptation of service compositions with variability models. Journal of Systems and Software, 91:24–47, 2014.	6
S08	J. Burdek, S. Lity, M. Lochau, M. Berens, U. Goltz, and A. Schurr. Staged configuration of dynamic software product lines with complex binding time constraints. In Proceedings of the Eighth International Workshop on Variability Modelling of Software-Intensive Systems, page 16. ACM, 2014.	4,25
S09	L. E. Sánchez, J. A. Diaz-Pace, A. Zunino, S. Moisan, and J.-P. Rigault. An approach for managing quality attributes at runtime using feature models. In Software Components, Architectures and Reuse (SBCARS), 2014 Eighth Brazilian Symposium on, pages 11–20.	3,75
S10	A. Murguzur, X. De Carlos, S. Trujillo, and G. Sagardui. Context-aware staged configuration of process variants@ runtime. In Advanced Information Systems Engineering, pages 241–255. Springer, 2014.	4,75
S11	K. Saller, M. Lochau, and I. Reimund. Context-aware dspls: model-based runtime adaptation for resource-constrained systems. In Proceedings of the 17th International Software Product Line Conference co-located workshops, ACM, 2013.	3,75
S12	C. Cetina, P. Giner, J. Fons, and V. Pelechano. Prototyping dynamic software product lines to evaluate run-time reconfigurations. Science of Computer Programming, 78(12):2399–2413, 2013.	4,25
S13	F. G. Marinho, R. M. Andrade, C. Werner, W. Viana, M. E. Maia, L. S. Rocha, E. Teixeira, J. B. Ferreira Filho, V. L. Dantas, F. Lima, et al. Mobiline: A nested software product line for the domain of mobile and context-aware applications. Science of Computer Programming, 78(12):2381–2398, 2013.	4,25
S14	I. Kumara, J. Han, A. Colman, T. Nguyen, and M. Kapuruge. Sharing with a difference: realizing service-based saas applications with runtime sharing and variation in dynamic software product lines. In Services Computing (SCC), 2013 IEEE International Conference on, pages 567–574.	3,25
S15	C. Ghezzi and A. M. Sharifloo. Dealing with non-functional requirements for adaptive systems via dynamic software product-lines. In Software Engineering for Self-Adaptive Systems II, pages 191–213. Springer, 2013.	5,25
S16	L. Jean-Baptiste, S. Maria-Teresa, G. Jean-Marie, and B. Antoine. Modeling dynamic adaptations using augmented feature models. In Proceedings of the 28th Annual ACM Symposium on Applied Computing, pages 1734–1741. ACM, 2013.	4,25
S17	H. Sabouri and R. Khosravi. Modeling and verification of reconfigurable actor families. JUCS, 19(2):207–232, 2013.	3,25
S18	D. Kramer, C. Sauer, and T. Roth-Berghofer. Towards explanation generation using feature models in software product lines. Knowledge Engineering and Software Engineering (KESE), page 13, 2013.	4,25
S19	V. T. Sarinho, A. L. Apolinario, and E. S. de Almeida. Oofm-a feature modeling approach to implement mpls and dspls. In 2012 IEEE 13th International Conference on Information Reuse & Integration (IRI), 2012.	2,25
S20	F. G. Marinho, P. H. Maia, R. Andrade, V. M. Vidal, P. A. Costa, and C. Werner. Safe adaptation in context-aware feature models. In Proceedings of the 4th International Workshop on Feature-Oriented Software Development, ACM, 2012.	4,25
S21	C. Parra, D. Romero, S. Mosser, R. Rouvoy, L. Duchien, and L. Seinturier. Using constraint-based optimization and variability to support continuous self-adaptation. Proceedings of the 27th ACM Symposium on Applied Computing, 2012.	3
S22	F. G. Marinho, R. Andrade, and C. Werner. A verification mechanism of feature models for mobile and context-aware software product lines. In Software Components, Architectures and Reuse (SBCARS), 2011 Fifth Brazilian Symposium on.	4,25
S23	C. Parra, X. Blanc, A. Cleve, and L. Duchien. Unifying design and runtime software adaptation using aspect models. Science of Computer Programming, 76(12):1247–1260, 2011.	3,75
S24	M. Rosenmuller, N. Siegmund, M. Pukall, and S. Apel. Tailoring dynamic software product lines. In ACM SIGPLAN Notices, volume 47, pages 3–12. ACM, 2011.	3,25
S25	J. Dehlinger and R. R. Lutz. Gaia–pl: a product line engineering approach for efficiently designing multiagent systems. ACM Transactions on Software Engineering and Methodology (TOSEM), 20(4):17, 2011.	4
S26	M. Acher, P. Collet, P. Lahire, S. Moisan, and J.-P. Rigault. Modeling variability from requirements to runtime. In Engineering of Complex Computer Systems (ICECCS), 2011 16th IEEE International Conference on, pages 77–86.	4,5
S27	L. Shen, X. Peng, J. Liu, and W. Zhao. Towards feature-oriented variability reconfiguration in dynamic software product lines. In Top Productivity through Software Reuse, pages 52–68. Springer, 2011.	3,75
S28	P. Fernandes, C. Werner, and E. Teixeira. An approach for feature modeling of context-aware software product line. J. UCS, 17(5):807–829, 2011.	4
S29	Z. Jaroucheh, X. Liu, and S. Smith. Mapping features to context information: Supporting context variability for context–aware pervasive applications. In Web Intelligence and Intelligent Agent Technology, 2010, International Conference on.	3,75
S30	Z. Jaroucheh, X. Liu, and S. Smith. Candel: product line based dynamic context management for pervasive applications. In Complex, Intelligent and Software Intensive Systems (CISIS), 2010 International Conference on, pages 209–216.	3
S31	T. Dinkelaker, R. Mitschke, K. Fetzer, and M. Mezini. A dynamic software product line approach using aspect models at runtime. In 5th Domain-Specific Aspect Languages Workshop, 2010.	4,5
S32	M. Acher, P. Collet, F. Fleurey, P. Lahire, S. Moisan, and J.-P. Rigault. Modeling context and dynamic adaptations with feature models. In 4th International Workshop Models@ run. time at Models 2009 (MRT'09), page 10, 2009.	3,25
S33	R. Wolfinger, S. Reiter, D. Dhungana, P. Grunbacher, and H. Prahofer. Supporting runtime system adaptation through product line engineering and plug-in techniques. In Composition-Based Software Systems, 2008, Seventh International.	3
S34	R. Capilla, M. Hinchey, and F. J. Daz. Collaborative context features for critical systems. In Proceedings of the Ninth International Workshop on Variability Modelling of Software-intensive Systems, page 43. ACM, 2015.	3,5
S35	J. C. Muñoz-Fernández, G. Tamura, I. Raicu, R. Mazo, and C. Salinesi. Refas: a ple approach for simulation of self-adaptive systems requirements. Proceedings of the 19th International Conference on Software Product Line, ACM, 2015.	4,25
S36	N. Abbas and J. Andersson. Harnessing variability in product-lines of self-adaptive software systems. In Proceedings of the 19th International Conference on Software Product Line, pages 191–200. ACM, 2015.	2,25
S37	R. Mazo, J. C. Muñoz-Fernández, L. Rincon, C. Salinesi, and G. Tamura. Variamos: an extensible tool for engineering (dynamic) product lines. Proceedings of the 19th International Conference on Software Product Line. ACM, 2015.	3,5

Modernizing the U.S. Army's Live Training Product Line using a Cloud Migration Strategy: Early Experiences, Current Challenges and Future Roadmap

Jeremy T. Lanman[1], Panagiotis K. Linos[2], LTC John Barry[3] and Amber Alston[4]
[1]U.S.A. Army, PEO STRI, Orlando, Florida, U.S.A.
[2]Butler University, Computer Science and Software Engineering, 4600 Sunset Avenue, Indianapolis, Indiana, U.S.A.
[3]U.S.A. Army, Modeling and Simulation, Ft. Eustis, Virginia, U.S.A.
[4]University of Central Florida, Computer Science, Orlando, Florida, U.S.A.

Keywords: Cloud Engineering, Training as a Service, Migration Strategy, Evolution, Testing, SOA.

Abstract: The integration of different networks, databases, standards, and interfaces in support of U.S. Army soldier training is an ever-evolving challenge. This challenge results in U.S. Army organizations repeatedly spending time and money to design and implement irreproducible architectures to accomplish common tasks. In response to this challenge, the U.S. Army has made significant improvements on its Live Training Transformation (LT2) product line to support the needs of live training simulations. Despite the progress with LT2 the Army continues to struggle to support the dynamic needs of the training units. These improvements have been inadequate due to growing technical complexities of interoperating legacy systems with emergent systems arising from advances in technology that suit the users' ever-changing needs. To better address and support the needs of the end-user, a cloud-based modernization strategy was crafted and deployed on the existing Common Training Instrumentation Architecture (CTIA). CTIA is the foundation architecture that provides software infrastructure and services to LT2 product applications. This paper describes some of the U.S. Army's initial experiences and challenges while crafting a cloud-based migration strategy to modernize its LT2 product line and underlining CTIA. It starts by providing some background and rationale and then it discusses the current state of this modernization effort followed by future directions including the U.S. Army's 2025 vision of its LT2 product line. The overall vision entails an evolution plan from today's standalone products to a modernized cloud-based TaaS (Training as a Service) approach. The Army's ultimate goal is to reduce complexity as well as operational and maintenance costs, while providing enhanced training for the Warfighter at the point of need, anytime, anywhere. Finally, this paper discusses some of the current challenges including the exploration of appropriate testing methodologies and related security issues for the SOA-based LT2 architecture and its services.

1 BACKGROUND

For over twenty years, the U.S. Army has addressed interoperability requirements between training systems through the creation and management of several system architectures. These systems are continually advancing in technology and growing in operational use in order to support the evolving needs of the U.S. Army's training communities. The Army has made significant strides in improving their current architectures, but these improvements have been inadequate to meet the growing training and system integration demands of users arising from technology advancement. Thus, it quickly became apparent that the need for a new architectural approach should be either developed or adapted in order to support the U.S. Army's live training environment.

1.1 Live Training

The U.S. Army uses many types of training simulations categorized as Live, Virtual, and Constructive (LVC). The focus of this paper is the architecture in support of the live simulation and training domain. Live simulation and training, defined by AR 350-1, is "real people operating real equipment" and is used to train and develop Soldiers' war-fighting skills (U.S. Army, 2011).

From an operations and training perspective, the surge in simulation and training technology and use was plagued by fragmentation and limited coordination between the U.S. Army branches due to

Lanman, J., Linos, P., Barry, L. and Alston, A.
Modernizing the U.S. Army's Live Training Product Line using a Cloud Migration Strategy: Early Experiences, Current Challenges and Future Roadmap.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 552-559
ISBN: 978-989-758-187-8

divergent operational demands, and the inability of technology to provide a "one size fits all" solution to the various needs of the operations and training community. This led to the consensus that limited interoperability was the highest level of integration possible at the time, which in turn led to the development of "stove-pipe" or "silo" systems across the Army's war-fighting functions: movement and maneuver, command and control, sustainment, protection, intelligence, and fires (M. G. Geruti, 2003)). The stove-piped systems were "able to send data to other applications within the same domain but not across boundaries" (R. L. Hobbs, 2003)). The impact of these stove-piped systems in live training ranges and instrumentation restricts their reusability, increases the cost to upgrade, and causes significant amount of range downtime to modify. In addition, the incompatibility between these disparate training systems results in the replacement of expensive components or requires the addition of adapters, which increases both cost and development time (M. Gomez, T. Kehr, 2011). The "stove-piped systems were built with different suites of sensors, networks, protocols, hardware, and software" (R. J. Noseworthy, 2010). The challenge and risk of linking stove-piped systems was identified in the 2006 Net-Centric Services Strategy that stated, "Patching stove-pipes together is a temporary solution; however, this leads to a fragile environment, which will eventually crumble under the high demands and unpredictable needs of the users" (DoD, 2006).

In the attempt to break down the barriers created by the stove-piped systems, the U.S. Army's Program Executive Office for Simulation, Training and Instrumentation (PEO STRI) developed the Live Training Transformation (LT2) product line to support the needs of live training. The first goal of the LT2 product line was to maximize commonality and systematic component reuse and to ensure interoperability across the live training community. The second goal was to reduce fielding time and acquisition cost and to provide "total ownership cost reductions across the live training domain" (J.T. Lanman et al, 2012). The LT2 product line supports home-station training, deployed training, Military Operations on Urban Terrain (MOUT) training, Maneuver Combat Training Center (MCTC) training and instrumented live-fire range training. Figure 1 depicts the initial three use cases for the live training architectural migration. The first use case entails the use of smart phones during combat training to capture soldier situational awareness and/or other related data and broadcast them to the command post data center for strategic decision making. The second use case presents various Army combat vehicles and soldiers transmitting training instrumentation data through defense or commercial satellite and network gateways back to the command post data center. Finally a future use case shows sharing and sending of information using special sensor devices connected via Bluetooth or other commercially available standards to various edge devices (e.g. tablet or smart phone) back to the command post data center for analysis.

1.2 Current Architectural Approach

The architecture that supports live training today is the Common Training Instrumentation Architecture (CTIA). CTIA is the foundation architecture of the LT2 product line that was developed by PEO STRI to specifically support live training. The benefits of CTIA have been seen in the reduction of development costs, sustainment costs, maintenance costs, and fielding time of live training ranges (J. T. Lanman et al, 2012). CTIA consists of architecture services, software components, standards and protocols that are Information Assurance (IA) certified to operate at the secret level (U.S. Army, 2013).

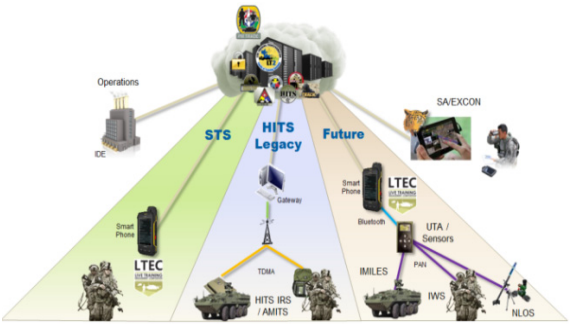


Figure 1: Initial use cases for live training.

Although CTIA has enabled units to conduct training through the benefits discussed in the previous section, CTIA technology is reaching its limitations to support the LT2 product line and ultimately the needs of the training community. This inability to evolve with user requirements has left a gap in supporting web interfaces and wireless mobile devices. This gap in support can be linked back to a lack of LT2 architectural vision that ties standards together resulting in live training components being highly dependent on the CTIA versions and limited backwards compatibility (J. T. Lanman et al, 2012). Additional challenges with CTIA include compatibility with other military systems, supporting distributed training center support, and scalability of footprint across LT2 product line. To address these

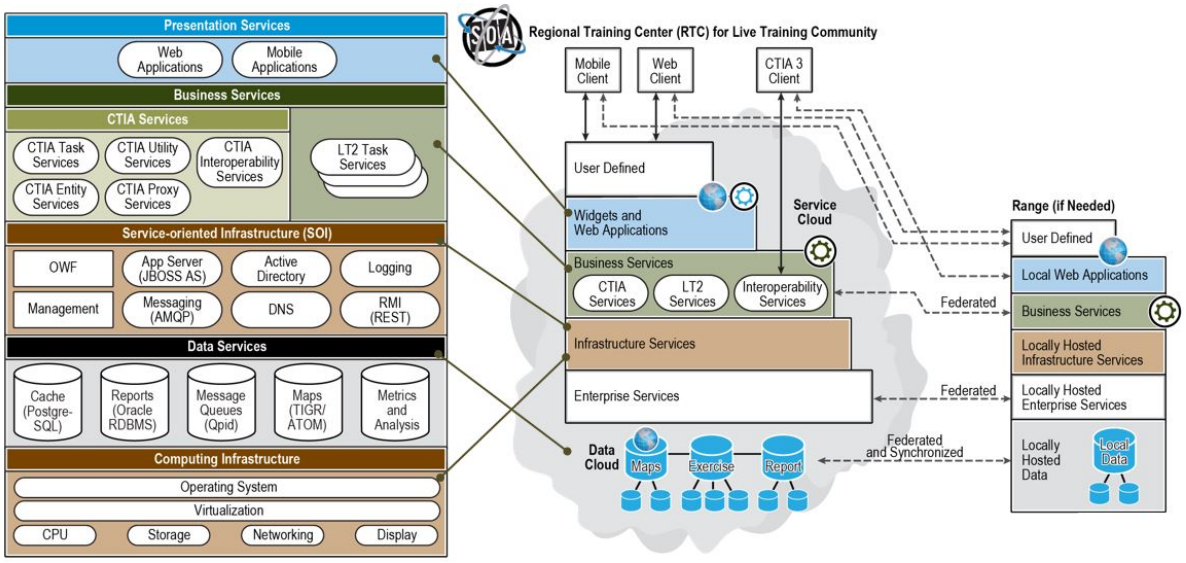


Figure 2: Conceptual view of a SOA-based cloud migration strategy for CTIA.

challenges, the architecture team and PEO STRI have adopted a Service-Oriented Architecture (SOA) migration strategy (J.T. Lanman et al, 2011). Figure 2 depicts a conceptual view of such strategy including a list of architectural layers and corresponding services provided. The architectural layers are mapped to the service layers of a conceptual Regional Training Center (RTC), a cloud-based data center that hosts services and data for training capabilities at various Army installations and ranges. If an Army range does not have a communications network, then a private mini RTC can be deployed at the range and later federated back to the main RTC data center.

1.3 Training as a Service

The term "Training as a Service" (TaaS) is used by the U.S. Army in order to refer to an "on-demand training environment" delivery model in which training software and its associated data are hosted in a cloud and are accessed by users using a thin client, normally using a web browser over the Internet.

The U.S. Army has deployed a TaaS strategy in order to develop simulation and training services (i.e. Web services) and the supporting infrastructure (i.e. networks, communications, sensors and computing hardware). Moreover, such TaaS strategy aims at building functional components and the supporting intermediate infrastructure according to modern cloud engineering principles and practices. It decomposes the system into components and layers. To obtain maximum flexibility and the greatest opportunity for reuse, each component exposes its capability through services available to the end-user and to other applications on the Army's Enterprise Network (AEN). By designing software around a set of services rather than a set of applications, TaaS aligns with the DoD migration to net-centricity and architectural patterns used in industry (DoD, 2012). Moreover, the architecture segregates the software that exposes persistent information (data services) from functional (or business logic) and presentation services. Both TaaS and the CTIA SOA and cloud infrastructure are built upon layered architecture frameworks. TaaS and CTIA SOA embraces consistent SOA and cloud-based concepts and architectural tenets, but differs in the sense that CTIA SOA is focused on defining architectural patterns that, while consistent with the TaaS objective architecture, focus on the unique issues of the instrumentation training environment rather than the holistic enterprise environment.

TaaS is now evolving while building common Army training apps and software services for Web browsers, desktop computers and mobile devices in the cloud environment. Army units and individual soldiers can access software applications such as a GPS tracking app for land navigation and exercise-control monitoring, tactical engagement simulation apps for laser and simulated fire engagements, and instrumented range apps for fixed live-fire targets. TaaS will eventually support up to brigade and battalion level force-on-force instrumentation and home station training with constructive simulation data feeds and battle damage assessment. TaaS is cloud-based with a deployable software service infrastructure to support the full live training domain. Figure 3 illustrates the LT2 based CTIA domains (home station, force-on-force, force-on-target)

supported by TaaS.

It is expected that CTIA will eventually support fully the mobile computing world. One of the goals here is to enable trainers to use mobile devices to capture training observations and evidence just like one might use an app to post a photograph to a social networking site. Finally, a more detailed description of how the Training as a Service (TaaS) delivery model is deployed by the U.S. Army can be found in (J. T. Lanman, P. Linos, 2013).

1.4 LT2 Vision

Figure 4 below summarizes the steps currently taken to fulfill the US Army's 2025 modernization vision of its LT2 product line and related CTIA. As depicted in Figure 4, the CTIA is migrated to a cloud-based architecture and all related products are converted to SOA services. This effort enables a transition from the standalone model to a target cloud-ready model, which will eventually reduce costs.

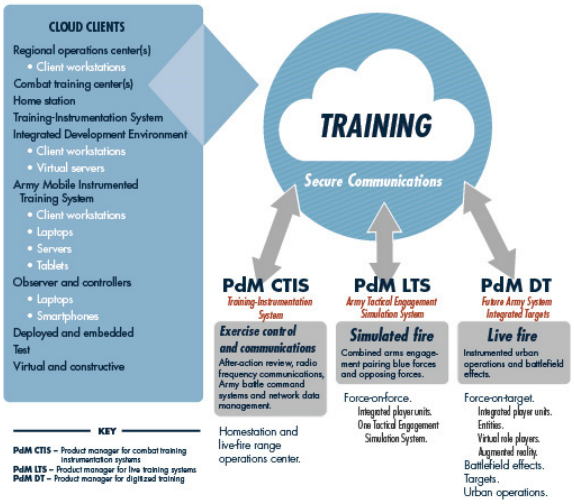


Figure 3: Training as a Service (TaaS) supporting the LT2 based CTIA domains.

In addition, all proprietary instrumentation systems are moved to a robust family of standards based sensor components. Also, the complete enterprise is modernized to a stand-up persistent, cloud-based LT2 enterprise of training services. Finally, the overall user experience is now moved from an operator-driven approach to a self-serve web or mobile apps tactic.

As part of meeting the LT2 vision objectives, Lanman and Linos discuss how decisions to change a product must be driven by the goals and objectives of the customer and other key stakeholders if they are to be successful (J. T. Lanman, P. Linos, 2012). Just as

Thrust	Description	Benefit(s)
Core SW	Move CTIA and Products to SOA services	• Reduce Costs (80% SLOC reduction) • Enable the Cloud model
Products	Move from standalone model to Cloud-ready	• Eliminate HW dependencies • Enable RTC concept
Instrumentation	Move from proprietary systems to a robust family of standards based sensor components	• Commoditize Instrumentation • Control evolution of capabilities • Leverage COTS
Enterprise	Stand-up persistent, Cloud-based LT2 Enterprise of Training Services	• Reduce Costs • Simplify Operations • Centralize SW, IA, Support
User Experience	Move from Operator-driven to Self-Serve Web / Mobile Apps	• Simplify Operations • Enhance Training • Capability Agile

Figure 4: A plan included in the U.S. Army's 2015 vision to modernize its LT2 product line.

the decision to adopt a product line approach for LT2 involves recognition of avoidable duplication, the decision to migrate to a cloud-based platform involves recognition of deficiencies in meeting upcoming fielding needs for CTIA based training systems. To ensure that the adoption of a cloud-based migration strategy addresses the true needs of the LT2 community, the architecture team, comprising key stakeholders based on influence and interest, have carefully defined and prioritized the strategic business goals and objectives for the LT2 architecture. CTIA provides the foundation for this architecture; however, business goals and objectives were extended to the LT2 community at large to ensure that the CTIA architecture aligns with community needs. These goals were then used to determine the priorities for the technology insertion effort.

1.5 Migration Roadmap

The migration roadmap of the current CTIA, to a modernized state, entails six Transition Architecture (TA) path-points (i.e. TA1 through TA6) as shown in Figure 5. Each such TA is based on a specific use case for tracking soldiers in both individual and small units training. Services allocated to each TA instantiation enable progressive levels of product team adoption. In addition, product teams are able to orchestrate the architecture services to meet their intended training use case, and develop user level application interfaces. Finally, each such transition architecture supports integration with first generation CTIA to the extent of the services provided. It is worth mentioning that TA6 was added to the migration plan later due to a funding opportunity with an LT2 product. That opportunity allowed the SOA to mature more quickly with additional services; however, it pushed the migration of other services out one year. The derived benefit was a faster deployment of a critical training capability at a major Army installation.

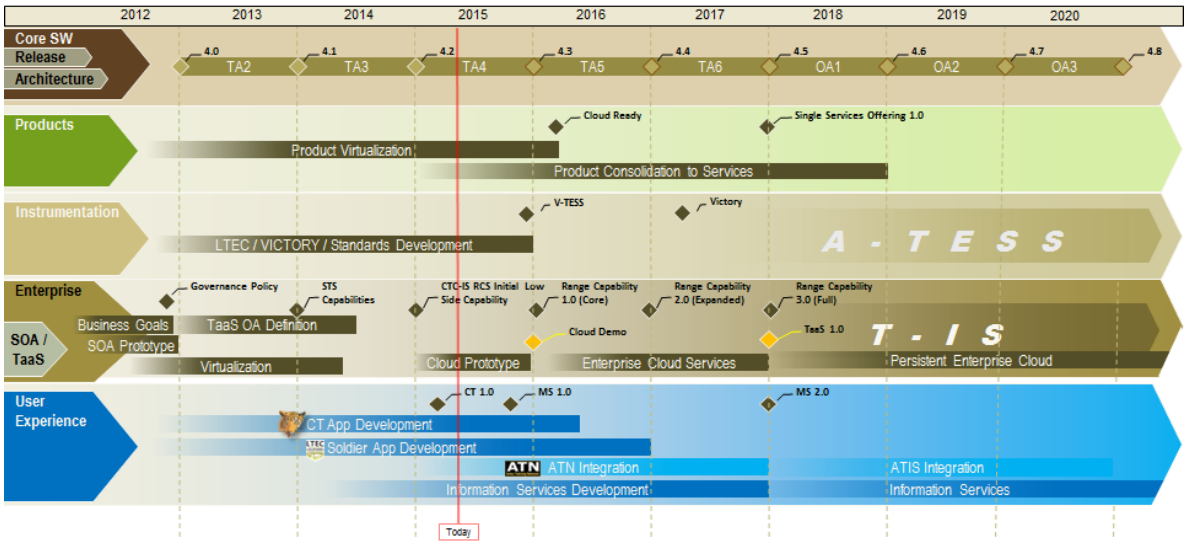


Figure 5: A roadmap depicting the evolution of Transition Architectures with related timeline.

The adopted migration roadmap entails modern cloud computing and virtualization technologies, which ensure effective interoperability among the Live, Virtual and Constructive (LVC) training systems and related applications. In addition, typical cloud engineering principles have been utilized while developing and orchestrating reusable, highly cohesive and loosely coupled software services at various granularity levels. Figure 5 also depicts the timeline and evolution of transition architectures TA1 through TA6 and their impact on core software, products, instrumentation, enterprise, and user experience.

As of today, transition architectures TA1, TA2 and TA3 have been already released with reasonable success. The future architectures will be evolving over the next ten years into a series of the remaining transition architectures (i.e. TA4 through TA6). More specifically, TA4 will provide services to support basic force-on-force instrumentation for brigade level home station training with constructive data feeds and battle damage assessment. Services will include asset tracking and exercise replay. Finally, TA5 and TA6 will be the last instantiation of the migration effort. Afterwards, the architecture transitions into production and sustainment for objective architectures OA1, OA2 and OA3. The target solution architecture (a.k.a. Objective Architecture) will be fully cloud-based with a deployable Service-oriented Infrastructure (SOI) supporting the full live training domain. Training will include up to battalion level force-on-force exercises integrating with mission command systems and entail full wrap-around live, virtual, and constructive interoperability capability.

2 CHALLENGES

2.1 Current Testing Issues

Changing to a cloud business methodology to support military requirements is unique because it changes the paradigm of traditional testing methodologies. This paradigm shift brings a new challenge of testing the architecture prior to implementation.

The current testing practice lacks the necessary metrics to validate the LT2's transition to a service-oriented capability. The current method is to "implement once ready" without putting the new or modified application through the rigors of a comprehensive testing framework. The current testing must expand from a solely integration focus to examining the effects of any new or modified application, such as interoperability and composability. Ignoring these effects may result in an open architecture without boundaries creating a governance nightmare.

In addition, the length of the potential migration amplifies this challenge. Currently, the new LT2 CTIA Objective Architecture Vision is a multi-year plan requiring that the SOA evaluation framework to be highly reusable and flexible. These characteristics allow the framework to adapt as end-user requirements naturally evolve over time as technology advances. With the long-term goal of developing testing criteria for the on-going evaluation of the LT2 architecture components, a thorough literature review is being conducted identifying potential applicable and validated SOA testing models that could support the LT2 cloud migration.

2.2 Observations on Cloud-based Testing Methods

Any reusable testing framework, including those utilized outside of the immediate LT2 architectural focus, must support the testing, data collection, and effective evaluation of the dependability and quality of services produced in a cloud-based services environment. Our literature review so far reveals various approaches and techniques for testing cloud-based applications at different test levels. Although these approaches and techniques seem promising, they lack the maturity of a validated and repeatable testing framework required to adequately assess the LT2 SOA-based implementation.

According to research conducted by Petrova-Antonova, et al., in support of their work to propose a possible SOA testing framework, there are several isolated testing tools and highly complex frameworks used in proprietary fashion for web service composition testing, however an available, proven, complete solution to both test and validate a SOA approach is still missing (D. Petrova et al, 2012).

Much of the challenge in developing a highly useful and repeatable testing framework is that cloud-based evaluation can be a highly complex computing issue. Services are dispersed over different deployment configurations; they must be highly adaptive as new services are added without requiring high levels of regression analysis. They may also be highly complex services with specific functionality offering differing operational tasks making on-going automation testing highly difficult (Y. Basili, 2012).

Youssef Bassil offers, as a part of his SOA-based framework development research, an overview of the five levels of evaluation that should be considered in any application of a SOA testing framework (Y. Bassil, 2012). They are as follows: *Unit Testing* (evaluating the individual service as an isolated element), *Integration Testing* (evaluating the SOA as a working group of co-joined services), *Regression Testing* (re-evaluating any recent updates to individual services across the working group of services), *Functional Testing* (evaluating that a service performs its intended purpose), *Non-Functional Testing* (evaluating properties such as availability and security vulnerabilities within the service).

In addition, Papastergiou and Polemi suggest that a proper testing framework used to evaluate a SOA approach to overcome interoperability challenges must include the following elements in order to confront recognized weaknesses in many of the currently available testing frameworks (S. Papastegiou, D. Polemi, 2010): *Clarified* (framework requires that testing apparatuses and necessary information are clearly defined as is the component being evaluated), *Adaptable & Extensible* (framework requires that new testing tools and methods, as well as new services, are easily integrated), *Flexible* (framework requires elements to be able to be adopted as needed for specific test cases, *Structured* (framework follows a concrete set of evaluation steps), *Interdependent & Scalable* (framework provides value independent of any given testing technology or number of services under test).

Although we identified several framework suggestions for testing a SOA-based implementation, there is clearly not a one-size-fits-all methodology to measuring the success of an implementation. Each evaluation framework must be informed by an understanding of the individual system needs and capabilities. The Army's live training systems, in all of their architectural complexity, particularly emphasize this need to take a customized approach of designing any intended evaluation framework with the LT2 goals and metrics of success top-of-mind. Therefore, we are still looking for experiences and recommendations on how to properly validate a SOA implementation (especially in the cloud) to be highly diverse. We will continue our literature review of related publications such as the one in (S. Tilley, T. Parveen, 2012).

3 LESSONS LEARNED

This section discusses lessons learned to date based on the on-going cloud-based modernization activities for the LT2 product line.

3.1 Leveraging Reuse

A lesson that we learned quickly during the cloud migration process is that common architecture frameworks succeed by providing a uniform and highly reusable feature-rich environment that allows developers to focus on their primary objective of implementing business-level use cases and not on repetitive implementation details. The Army's success with CTIA can be realized by the fact that it forms an average of 50% of the code base for all live training systems deployed since 2006. More specifically, the Army's LT2 products typically use about 57% of an approximate two million lines of code in the CTIA framework (J. T. Lanman et al, 2012). Another realization is that due to the large investment of the current architecture and the multitude of component dependencies it is unreasonable to expect that the new architecture can be developed in an isolated environment and deployed to replace completely the

existing architecture. Therefore, it became apparent that backwards compatibility must be maintained with legacy software components through the existing CTIA framework interfaces. For more information on the historical evolution of CTIA we refer the reader to (J. T. Lanman, P. Linos).

3.2 Mapping Business Processes

In the live training domain the technical problem does not directly map to the IT business process for producing goods and services which SOA is typically modeled upon. As we know, the standard SOA business process is an orchestration of multiple business functions each of which rely on the results of the previous function to accomplish a discrete task. For instance, consider the archetypal example where a business process entails: booking an order => updating inventory => shipping => billing. In the case of a live training environment, business units are providers of content and context to artifacts generated within the system. The system collects artifacts and then consumers generate review content from the artifacts. The path through artifact generation, manipulation and presentation is not dictated by a predefined orchestration but by an ad hoc manner, depending on the fidelity of the training environment. A combat training center for example has multiple organizations dedicated to providing context and content for discrete aspects of the training exercise such as fires, upper echelon support, or aerial support, whereas a home station training range instantiation is typically an individual assessing the time on target, or efficiency in meeting the training objectives for a single unit. As a result, the lesson here is that the U.S. Army could not directly adopt a traditional IT business process, but customized such process to allow for scalability of multiple or simultaneous training assets to accomplish discrete tasks in a traditional SOA implementation.

3.3 Improving Deployment Time

In an archetypal cloud-based deployment the SOA system is ubiquitous and accessible from multiple disparate organizations. In addition, the system is always available with no defined end state. In the case of the U.S. Army's live training systems however, as they are deployed currently, installations exist as isolated standalone systems. Also, training exercises have specific training objectives and they access data that are not shared between concurrent exercises at different ranges. Moreover, service composition is a function of the training audience and range, from combat training centers with dedicated rack servers down to individual ranges consisting of a single workstation operated directly by an individual from the training unit. These systems are accessed and maintained onsite and their state is dependent on the phase of the training rotation. Although the target migration architecture (a.k.a. Objective Architecture) will be a ubiquitous solution overall, the cloud-based implementation of CTIA must account for the different phases of training where different subsets of services are available depending on the training rotation state. As a result, we have learned so far that additional architectural layers and specialized federation services are needed in the Objective Architecture in order to account for the various asynchronous training phases.

3.4 Assuring Security

Any changes to the CTIA and LT2 architectures and their components must consider the security and accreditation impact in order to comply with the Information Assurance (IA) policies and the DoD's Risk Management Framework (RMF) process. We need also to consider that as the U.S. Army evolves and in order to implement cloud computing and virtualization, that the security and IA requirements are also likely to evolve and introduce new requirements. Therefore, the lesson that we derived from this is to engage with security experts early in the cloud services design process and build in security constructs in the design, which allows for easier IA certification and a more secure and cost effective solution.

3.5 Pending Technical Concerns

We found that cloud-based migration challenges mostly concentrate around bandwidth, scalability, and technical issues based on SOA related implementation details such as limitations of proprietary Enterprise Service Bus (ESB) capabilities. Historical CTIA development has resulted in a set of metrics that ensure that all development activities comply with the required Technical Performance Measures (TPMs), which are internally defined by the U.S. Army. Continuous integration and testing ensures that any time these metric values are exceeded the development team takes immediate action. Also, existing test harnesses and training scenarios provide a baseline for validation testing. It is worth mentioning that the current system's performance exceeds the performance of the previous generation of CTIA. The first transition architectures focus on the most reused and also the most

performance-sensitive elements within the system, ensuring that these issues are addressed early and often. The CTIA SOA is early in its development and we are still collecting metrics on performance and testing. However, since these technical concerns are an on-going investigation, we will continue to identify and address them as needed. We plan to include those findings and related data in future reports.

4 CONCLUSIONS

In this paper, we briefly described the U.S. Army's overall effort, experiences and lessons learned while modernizing its Live Training Transformation (LT2) product line. To this end, the U.S. Army decided to leverage the industry-wide knowledge and success of cloud computing using SOA, and it has identified this business approach as the new migration to its LT2 product line. Based on work accomplished so far from such an effort, the U.S. Army believes that this transition is already increasing the interoperability of its different networks, databases, and interfaces that support live training. At the same time, the U.S. Army also acknowledges the fact that this paradigm shift poses various new challenges. For instance, just as there is not an "out of box" cloud-based strategy, there is not a "one size fits all" testing framework. Based on such an observation we have found and discussed above some existing cloud-based testing techniques and approaches that could be used for the LT2 transition. However, we understand that further investigation is needed here before any decisions are made.

We have also made an attempt in this paper to explain how the U.S. Army's PEO STRI has incorporated the concept of TaaS (Training as a Service) in order to successfully migrate and modernize its simulation and training legacy software for the live training domain. Based on early observations, it appears that the TaaS strategy addresses the need to reduce costs and leverage technology developments in order to better support the soldiers' training needs. However, as we have described in our lessons learned section above, there exist some challenges. For example, SOA and cloud adoption related caveats are typically centered on network bandwidth, latency, software scalability and other technical issues. Furthermore, any changes to architectures and software services must consider the security and accreditation impacts that might affect information assurance.

REFERENCES

U.S. Army, AR 350-1: Army Training and Leader Development, Washington, DC: United States Army, 2011.

M. G. Ceruti, "Data Management Challenges and Development for Military Information Systems," *IEEE Transaction on Knowledge and Data Engineering,* pp. 1059-1068, 2003.

R. L. Hobbs, "Using XML to Support Military Decision-Marking," in *In Proceedings of the 2003 Winter Simulation Interoperability Workshop*, Orlando, 2003.

M. Gomez and T. Kehr, "Leveraging Service-Oriented Architectures (SOA) within Live Training: An Assessment," in *Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC)* , Orlando, 2011.

R. J. Noseworthy, "Supporting the Decentralized Development of Large-Scale Distributed Realtime LVC Simulations Systems with TENA (The Test and Training Enabling Architecture)," *14th IEEE/ACM Symposium on Distributed Simulation and Real-Time Applications,* pp. 21-29, 2010.

DoD, CIO, *Net-Centric Environment to an Enterprise Service Oriented Architecture,* 2006.

U. S. Army PEO STRI, "Live Training Transformation (LT2) Product Line," [Online]. Available: http://www.peostri.army.mil/PM-TRADE/lt2_productline.jsp. [Accessed 22 March 2013].

J. T. Lanman, S. R. Clarke, S. Darbin Hillis and D. Frank, "Applying Service Orientation to the U.S. Army's Common Training," in *Iterservice/Industry Training, Simulation, and Education Confernce (I/ITSEC) 2012*, Orlando, 2012.

J. T. Lanman, S. Clarke, R. Darbin and D. Frank, "Applying Service Orientation to the U.S. Army's Common Training Instrumentation Architecture," in *Interservice/Industry Training, Simulation and Education Conference*, Orlando, 2012.

J. T. Lanman, S. Horvath and P. Linos, "Next Generation of Distributed Training utilizing SOA, Cloud Computing and Virtualization," in *Interservice/Industry Training, SImulation and Education Conference*, Orlando, 2011.

DoD, "Cloud Computing Strategy," Chief Information Officer, 2012.

J. T. Lanman and P. Linos, "Employing Service Orientation to Enable Training as a Service (TaaS) in the U.S. Army," in *International Conference on Cloud Engineering (IC2E)*, San Francisco, 2013.

D. Petrova-Antonova, S. Illieva, I. Manova and D. Manova, "Towards Automation Design Time Testing of Web Service Compositions," *e-Informatica Software Engineering Journal,* vol. 6, no. 1, pp. 61-70, 2012.

Y. Bassil, "Distributed, Cross-Platform, and Regression Testing Arhitecture for Service-Oriented Architecture," *Advances in Computer Science and its Applications (ACSA),* vol. 1, no. 1, 2012.

S. Papastergiou and D. Polemi, "A Testing Process for Interoperability and Conformance of Secure Web Services," in *Radio Communications*, A. Bazzi, Ed., Rijeka, Croatia, InTech, 2010, pp. 689-712.

S. Tilley and T. Parveen, Software Testing in the Cloud: Migration and Execution, Springler, 2012.

Aligning Software Design with Development Team Expertise

Jānis Grabis[1], Egils Meiers[2], Inese Šūpulniece[1], Solvita Bērziša[1], Edgars Ozoliņš[2] and Ansis Svaža[2]
[1]*Institute of Information Technology, Riga Technical University, Kalku 1, Riga, LV-1658, Latvia*
[2]*Visma Enterprise, Kronvalda bulv.3/5, Riga, LV-1010, Latvia*
{grabis, inese.supulniece, solvita.berzisa}@rtu.lv, {egils.meiers, edgars.ozolins, ansis.svaza}@visma.lv

Keywords: Enterprise Application, Refactoring, Development Reorganization, Team Expertise, Clustering.

Abstract: Large enterprise applications are developed by teams of developers specializing in particular functional or technical areas. An overall application architecture is used to guide allocation of development tasks to the development teams. However, quality of the architecture degrades over the application life-cycle and manual refactoring is challenging due to the size and complexity of enterprise applications. This paper proposes to use automated clustering of large enterprise applications, where clusters are built around application business centers, as a means for refactoring the software design with an objective to improve allocation of software modules to development teams. The paper outlines a module allocation process in the framework of the overall enterprise application development process and reports an illustration of the allocation process. The illustration is based on the case of refactoring of a large third tier ERP system.

1 INTRODUCTION

Development of large software applications such as Enterprise Resource Planning (ERP) systems is a complex task. These systems are constantly evolving and huge efforts are devoted towards maintenance of existing applications and developing new functionality. Expertise of development team is a crucial factor to ensure efficient maintenance and software evolution (Bennett and Rajlich, 2000). That is especially important for large multi-functional applications because for their wide scope and long life-cycles. Developers specialize in particular functional and technical areas to ensure development efficiency (Liang, 2010). This specialization is enabled by having a modular system design (Paulish, 2002). Unfortunately, the system design if initially present tends to deteriorate during the life-cycle for large complex applications (Cai et al., 2009).

This paper investigates a problem of refactoring the system design of long life-cycle packaged applications with an objective to support modularized development by dedicated teams. The refactoring is achieved by automated clustering of the system into self-contained modules. The automated clustering is considered because manual refactoring is prohibitive in the case of large systems. It is assumed that development of the modules requires specific development expertise and teams are formed and the modules are assigned to them to attain the best match between the required competencies and the team's expertise.

The objective of this paper is to propose a method for aligning software design and team's expertise. The method is geared towards development of packaged applications including ERP systems. ERP development is investigated from the vendor perspective (as opposed to the ERP implementation perspective). Application of the method is illustrated using an example of the third tier ERP system undergoing a system's redesign project. The further research is intended to focus on evaluation of actual benefits of redesigning of the ERP systems from the vendor's perspective what is an insufficiently exposed research and practical problem. The main expected contribution of the proposed research is to determine suitability of automated refactoring to guide development team assignment and to facilitate inter-team collaboration in the case of large-scale packaged applications.

The rest of the paper is organized as follows. Section 2 describes the ERP development process highlighting its modular nature and discusses the role of development team's composition. Section 3 introduces a process for allocating modules to development teams. Section 4 describes preliminary evaluation of the alignment process. Section 5 concludes.

Grabis, J., Meiers, E., Šūpulniece, I., Bērziša, S., Ozoliņš, E. and Svaža, A.
Aligning Software Design with Development Team Expertise.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 560-565
ISBN: 978-989-758-187-8

2 ERP DEVELOPMENT PROCESS

An ERP development process resembles the traditional software development process. Two distinguishing features of this process are specific aspects of requirements management and wide scope of the application resulting in functional and technological complexities. Monnerat et al. (2008) suggest to use enterprise modeling techniques to establish a comprehensive set of requirements covering all areas of application of ERP systems. The incremental approach (Sommerville, 2010) to evolving functionality of the ERP systems on the basis of key requirements and overall architecture is used to address the functional and technological complexities. Figure 1 shows an overall ERP development process.

An ERP system can be developed from scratch or by evolving existing software. The latter case is more common in practice since either the previous version of the ERP system is available or the ERP system development is a continuation of successful custom software development. In this research, we focus on maintenance and evolution of existing ERP systems. The development process is driven by feedback from customers, market trends, changes in regulatory requirements and other factors (Xu and Brinkkemper, 2007). The enterprise modeling activity concerns scoping of ERP development and identification of key requirements towards the ERP system. ERP systems consist of functional modules, which cover certain areas of enterprise activities. Modules can be developed relatively independently (modules from the development perspective are not necessarily the same as modules from the functional perspective). However, to ensure development and usage efficiency and consistency, the functional modules are developed following common principles determined according to the base requirements and operationalized in the overall architecture or systems design. The individual modules are integrated together in order to release a new version of the ERP systems to customers. The module development, integration and release are continuous processes, especially, if agile techniques are used in development (De Carvalho et al., 2010).

Enterprise modeling requires participation of process owners and key users (Sandkuhl et al., 2014). They specialize in different business areas of the enterprise and possess limited knowledge and understanding about specific aspects of other business areas. Moreover, the research suggests that cross-functional teams have negative impact on implementation of ERP systems (Lui and Chan 2008). Similarly, agile development practices suggest using vertical teams rather than horizontal teams (Ratner and Harvey, 2011). Carmel and Bird (1997) provide evidence that packaged systems are usually developed by teams of up to five developers. Therefore, it is often practical and advisable to distribute ERP development activities among teams specializing in particular business areas.

Software architecture plays a major role in dividing software into manageable modules assigned to individuals or small teams for development (Unphon and Dittrich, 2010). However, that might be hampered by intricacies of the ERP technical design (Rettig, 2007), i.e. ERP systems consist of a large number of components linked together in a complex web of associations, which has evolved during the life-cycle. The overall architecture can be improved by refactoring although manual refactoring of large enterprise applications is challenging. This paper explores automated decomposition of ERP systems as a part of software design refactoring to improve allocation of modules to development teams.

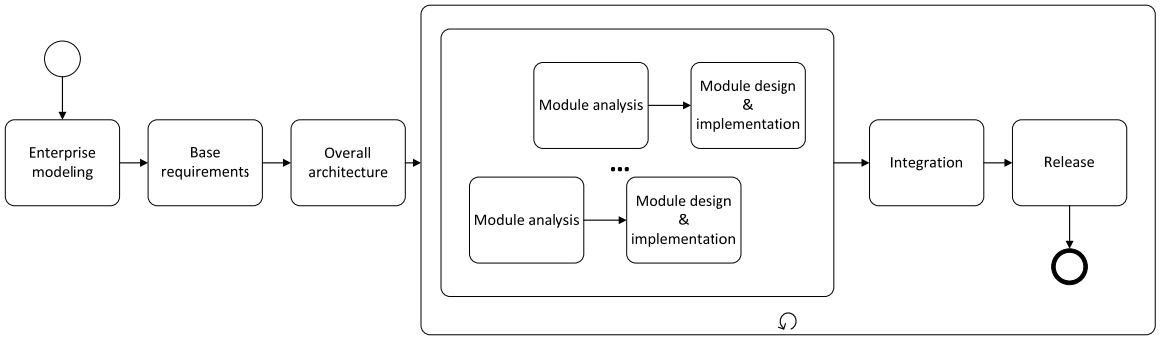


Figure 1: ERP systems development process.

3 ALIGNMENT APPROACH

The alignment approach is elaborated for an ERP system which requires major architecture refactoring. The refactored architecture will be used to guide future software evolution processes including project management and team assignment processes.

System redesign and identification of modules takes place at certain milestones of software evolution. Development teams change more frequently. However, it is assumed that once a module has been assigned to team knowledge is preserved in it even though team members change occasionally. Alignment between team expertise and design also needs to be periodically updated since newly developed components are assigned to modules and characteristics of modules might change.

The system is divided in modules using clusters built around business centers (Figure 2). The business centers are system's design components identified by a system architect as being central to providing desired functionality. The clustering is performed automatically and clusters consist of closely related components as measured by strength of associations among the components. There are components having only internal associations within a cluster and there are associations spanning boundaries of the clusters. The latter associations are particularly important to determine interfaces and to set contracts among development teams. The clustering addresses just some of the system's redesign concerns. It is used as an input to other refactoring activities (e.g., Riva 2004), which yield the final division of the systems into modules. Competency requirements are identified for every module. They concern knowledge of specific functional or technical areas associated with a particular module. For instance, ab absence management module requires knowledge of human resources management.

Development teams work continuously throughout the system's life-cycle and has certain functional and technical competencies. The available competencies concern knowledge possessed by team members. Experience in a specific functional or technical area plays a major importance in determining team competencies. The modules are assigned to the development teams by matching the available competencies and the competency requirements. Some changes in teams' composition can be introduced to achieve a better match.

4 PRELIMINARY RESULTS

Feasibility of the alignment approach is evaluated by analyzing a third tier ERP system. This system is a multi-module system developed by its vendor over 20 years using object-oriented development techniques. The systems has about 4 million source lines of code, 26,000 classes containing business logics and about 160K associations. IT is a three-tier client-server system though architectural principles, system design and styles of programming as well as functional requirements have experienced many changes and maintenance and development of new functionality have become increasingly complicated. The company has initiated a system's redesign project. In order to simplify the system's design it is attempted to improve decomposition of the system in modules. Given the size of the system, at least initial decomposition is performed using automated clustering techniques. The improved decomposition is envisioned to facilitate assignment of development teams to individual modules of the systems.

The company has about 10 teams working on system's evolution. A team usually includes a product owner, business expert, two to five developers and a couple of tester depending on workload. The business expert represents customer needs and specializes in a particular technical or

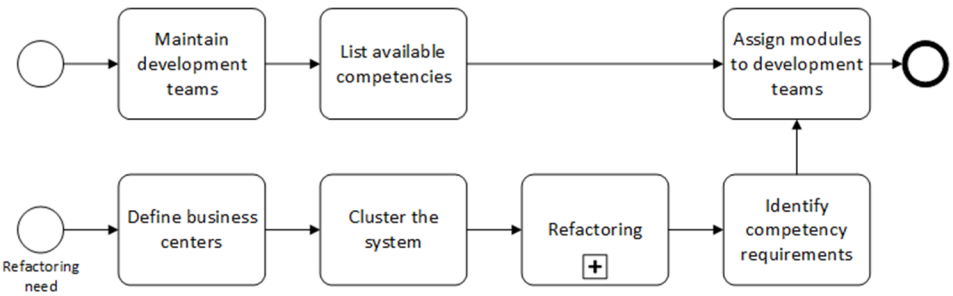


Figure 2: Alignment approach.

functional area. The product owner creates development tasks to implement the requirements. Successful product owners have intimate understanding of functional as well as technical aspects of her modules. Developers and testers have technical competencies and are more productive if they have sufficient understanding and experience about a given module.

45 to 50 tentative business centers are identified. For example, there is an industry specific solution for forest management, which has classes implementing functionality for forest clearances management, wood transportation and billing. Even though clusters are built around the business centers, new clusters also can emerge during the clustering process.

The ERP system is clustered using a hierarchical clustering algorithm (e.g., Cui and Chae 2011). The clustering is performed using a systems representation as a graph as an input. The graph's nodes are source code modules and classes. The graph's edges have several types including uses, extends, implements and other associations. Nodes are attached to clusters to maximize a similarity measure calculated as a weighted sum of edges connecting the node to candidate clusters. The technical description of the clustering algorithm is beyond the scope of this paper and additional details are provided in (Šūpulniece et al., 2015).

The clustering yields around 100 clusters though the right level of granularity is yet to be determined. Figure 3 shows a fragment of high level clustering results. Clusters are shown as bubbles and associations connect interrelated clusters. It can be observed that there is a relatively large number of inter-cluster associations even after the clustering and the clusters have varying degree of centrality.

Figure 4 zooms in on three clusters. The bubble size represents the number of intra-cluster components. Ovals surrounding a bubble and enclosed within a square indicate components having inter-cluster associations. These components are of particular interest because they will serve as interfaces among development teams. One of the clusters identified is a cluster for processing customer payments. This cluster has 229 intra-cluster components and 86 components interfacing with other clusters (there is more than a thousand intra-cluster associations).

The clustering results do not represent a ready-to-be-used new technical design of the system and are not directly transferable to development. It is possible that a single cluster might require different competences due to inefficiency in the current systems design. The clusters will be used by system architects and other stakeholders for discussions on redesigning the system. That will lead to a set of software modules, which could be assigned to individual development teams.

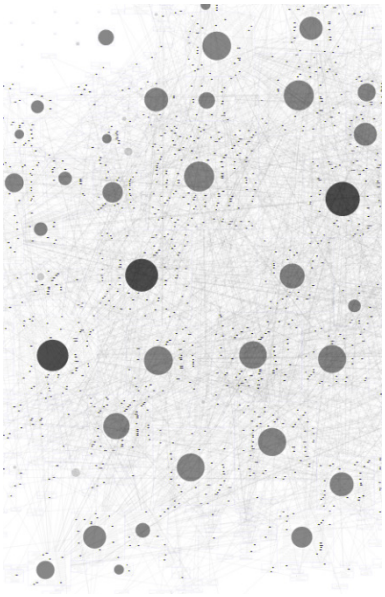


Figure 3: A fragment of clustering results.

Figure 5 illustrates allocation of modules to development teams. This illustration focuses on five tentative modules: 1) financial accounting (FA) billing; 2) sales and distribution (SD) sales order processing; 3) forest management (FM) billing; 4) FM clearance; and 5) FM transportation. The former modules are cross-sectional, while the latter three modules belong to a horizontal solution developed specifically for the forestry industry. The identified competency requirements are given in Table 1 (the knowledge of the base development technologies applies to all modules.

Table 1: Competency requirements for tentative modules.

Module	Required competencies
FA billing	FA
SD sales order processing	CRM
FM billing	FA
FM clearance	FM, GIS integration
FM transportation	FA, GIS integration

Among the development teams, there are teams FA, customer relationships management (CRM) and forest management, respectively. Team FA has expertise in functional aspects of financial accounting what matches to the FA Billing module. Similarly, Team CRM specializes in customer facing

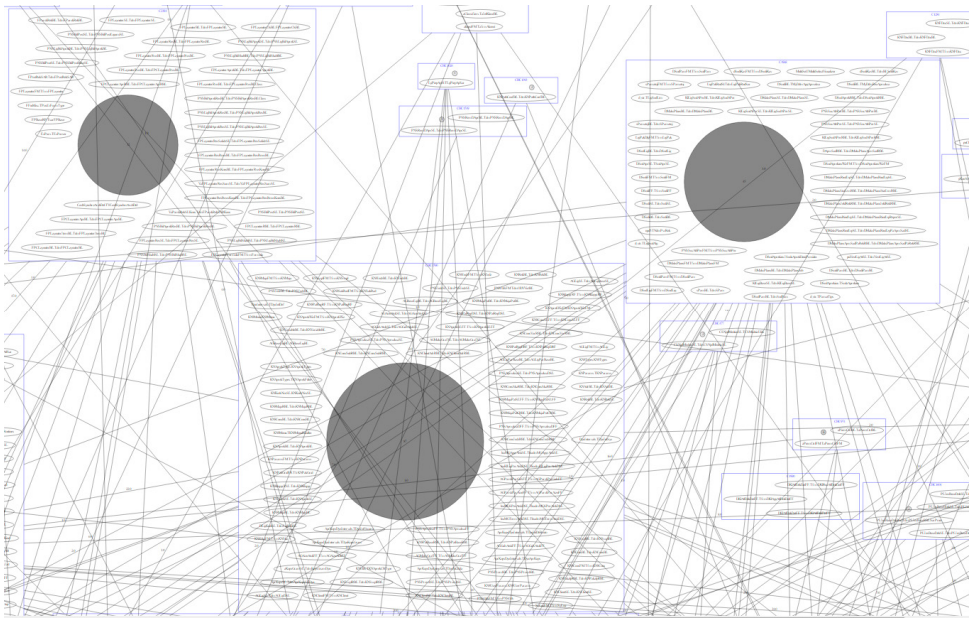


Figure 4: Sample clusters showing intra-cluster and interface components.

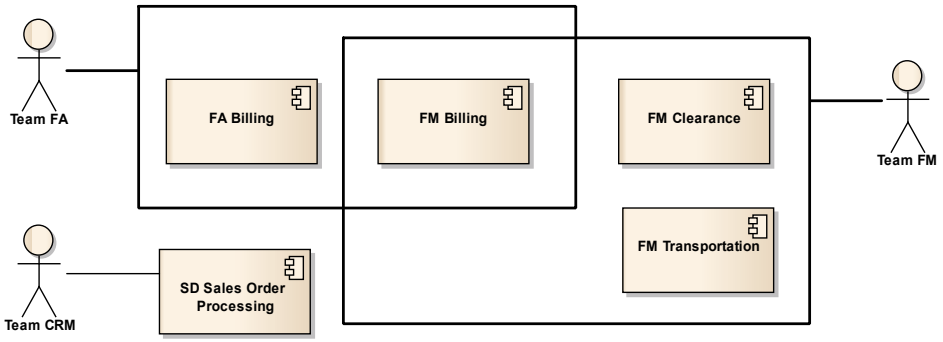


Figure 5: An illustrative matching between teams and modules.

processes what matches to the SD Sales order processing module. Team FM has experience in working with forest management related functionality. However, the FM Billing module also requires FA competencies and there is a decision to be made about allocating this module to one of the teams.

In many cases development teams can be rearranged to find the best fit between modules and teams. Otero et al. (2009) describes a formal approach for assigning teams according to their competencies. This method could be adopted for purposes of this investigation. It also accounts for varying degrees of competency and experience.

5 CONCLUSIONS

The paper proposes a method for automated clustering of enterprise applications as a means for allocating modules to development teams. It is argued that ERP systems are best developed by teams specializing in specific functional and technical areas. The overall architecture is used to allocate modules to these specialized development teams. Clustering is used for automated identification of the modules because manual refactoring is prohibitive. Business centers are used as a starting point of clustering to attain better alignment between software design and expertise of development teams.

The decomposition based allocation is expected to bring the following benefits: 1) teams can specialize in particular functional and technical areas of application development; 2) clear separation of responsibilities among the teams; and 3) faster integration testing (i.e., teams are responsible for

intra-module testing and integration testing focuses only on interface components). From the practical perspective, research results will be used to find the best allocation of modules to development teams and to manage collaboration among the teams. From the theoretical perspective, further research is expected to provide insights in ERP development from the vendor perspective and to evaluate actual benefits of software design refactoring.

There are several challenges to be addressed. The first challenge is finding the appropriate level of granularity or cluster size. The second challenge is definition of modules on the basis of clustering results. A special attention should be devoted to clusters mixing various expertise requirements and to identification of competency requirements for the modules. Finally, the module to team allocation method should be formalized. The granularity level will be determined in experimental studies and by receiving feedback from the development team. The modules will be developed by involving software architecting experts. The evaluation will be performed by means of the case study and comparative analysis of software development efficiency measures.

One of the main challenges is to convince development teams that automated refactoring suggests appropriate solutions for changing the long-established way of working and collaborating among the teams.

ACKNOWLEDGEMENTS

The research is funded by the ERDF project "Information and communication technologies competence center" Nr. KC/2.1.2.1.1/10/01/001 (Contract No. L-KC-11-0003, www.itkc.lv) activity 1.3. "The Method of Monolithic System Decomposition According to SOA Principles."

REFERENCES

Bennett, K.H., Rajlich, V.T., 2000. Software Maintenance and Evolution: a Roadmap. *Proceedings of the Conference on The Future of Software Engineering*, pp. 75-87.

Cai, Z., Yang, X., Wang, X., Wang, Y., 2009. A systematic approach for layered component identification. In 2009 2nd *IEEE International Conference on Computer Science and Information Technology*, pp. 98–103.

Carmel, E., Bird, B., 1997. Small is beautiful: a study of packaged software development teams, *Journal of High Technology Management Research*, 8(1), 129-148.

Cui, J.F., Chae, H.S., 2011. Applying Agglomerative Hierarchical Clustering Algorithms to Component Identification for Legacy Systems. Information and Software Technology, 53(6), 601-614.

De Carvalho, A., Johansson, B., Manhães, R.S., 2010. Agile software development for customizing ERPs. In *Enterprise Information Systems and Implementing IT Infrastructures: Challenges and Issues*, pp. 20-39.

Liang, T.P., Jiang, J., Klein, G.S., Liu, J.Y.C., 2010. Software Quality as Influenced by Informational Diversity, Task Conflict, and Learning in *Project Teams, IEEE Transactions on Engineering Management*, 57(3), 477-487.

Lui, K.M., Chan, K.C.C., 2008. Rescuing Troubled Software Projects by Team Transformation: A Case Study with an ERP Project. *IEEE Transactions on Engineering Management*, 55(1), 171-184.

Monnerat, R.M., De Carvalho, R.A., De Campos, R., 2008. Enterprise systems modeling: The ERP5 development process, *Proceedings of the ACM Symposium on Applied Computing*, pp. 1062.

Otero, L.D., Centeno, G., Ruiz-Torres, A.J., Otero C.E., 2009. A systematic approach for resource allocation in *software projects. Computers & Industrial Engineering 56*, 4, 1333–1339.

Paulish, D., 2002. *Architecture-Centric Software Project Management*. Addison-Wesley, Boston, MA, USA.

Ratner, I.M., Harvey, J., 2011. Vertical slicing: Smaller is better. *Proceedings - 2011 Agile Conference*, Agile 2011, pp. 240-245.

Rettig, C., 2007. The trouble with enterprise software. MIT Sloan Management Review 49(1), 21-27+90.

Riva, R., 2004. *View-based Software Architecture Reconstruction. PhD thesis*, Technical University of Vienna.

Sandkuhl, K., Stirna, J., Persson, A., Wisotzki, M., 2014. *Enterprise Modeling: Tackling Business Challenges with the 4EM Method*. Springer, Berlin.

Sommerville, I., 2010. *Software Engineering*. Person, 9th Edition.

Šūpulniece, I., Polaka, I., Bērziša, S., Ozoliņš, E., Palacis, E., Meiers, E., Grabis, J., 2015. Source Code Driven Enterprise Application Decomposition: Preliminary Evaluation. ICTE in Regional Development 2015 Valmiera, Latvia, Procedia Computer Science 77, pp. 167-175.

Xu, L., Brinkkemper, S., 2007. Concepts of product software. *European Journal of Information Systems. 16(5)*, pp. 531-541.

Unphon, H., Dittrich, Y., 2010. Software architecture awareness in long-term software product evolution. *Journal of Systems and Software 83(11)*, 2211-2226.

Our Orthodox Methods and Tools Are 100 Years Old and Due for Replacement

Ronald Stamper
Formerly University of Twente and London School of Economics, Now 38 London Court, 9-13 London Road, Oxford OX3 7SL, U.K.
stamper.measur@gmail.com

Keywords: Information, Analysis Methods, Specification Tools, Scientific Paradigms, Scientific Method, Refutationism, Law, Norms, Signs, Organisational Semiotics, Taylorism, Semantics, Affordances.

Abstract This paper is intentionally provocative. The analysis methods and specification tools we use today are derived from the century-old Taylorism via office work-study. If that was our scientific foundation, many obvious anomalies should have forced us to find a new paradigm. Rejecting *information-flow* in favour of a *knowledge-field* paradigm, we can build a rigorous science of organisational semiotics to underpin the engineering of information systems, taking account of the essentially human and social aspects of information: semantics/meaning, pragmatics/intention and social products/value, while reaching the level of rigorous formality needed for the technical aspects of the system. Practical case studies have demonstrated the advantages of this new approach, which reduces costs, especially over a long period while making the system easier for the users to understand.

1 INTRODUCTION

We need a sound scientific foundation for engineering organisational information systems that encompasses the organisational as well as the technical.

How do we compare? Hardware evolves phenomenally fast; software less so; and, 60 years on, AI still threatens, like Shakespeare's King Lear, to "do such things, what they are, yet I know not; but they shall be the terrors of the earth." As an example, Stephen Hawking told the BBC: "The development of full artificial intelligence could spell the end of the human race." But we are slower still. IS systems analysis and design clings to Taylor's 100-year-old scientific management. Today's UML, looks modern but it embodies the same old ideas.

1.2 Machines

UML, 1960s' ISAD tools and Taylor's 1890s work-study tools all track the flow of parts and materials and sequences of operations performed on them. Usually, in factories these are mechanical products, but in offices, documents and in computer systems, structured data. Taylor's science concerns only the movements of and operations upon objects and materials. So, importing his science into our domain limits 'information science' to some purely technical aspects and forces us to treat every organisation as a kind of machine.

Is that enough? Probably not!

1.3 Organisations

Back in the 1960s, the steel industry had an acute shortage of systems analysts, and they asked me to create courses to address the problem. Computer manufacturers providing the only other training at that time, taught how to introduce computers into a business. That technical bias and lack of understanding of the human and social aspects of information systems seemed to explain the alarming project failure rate. We should be equally alarmed today because the failure rate is still high.

1.4 Mystical Fluids

Instead, hoping to teach how to improve an *organisation* as an information system, using technology where appropriate, I searched for a scientific understanding of the role information plays in the functioning of organisations. To start with,

Stamper, R.
Our Orthodox Methods and Tools Are 100 Years Old and Due for Replacement.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 566-571
ISBN: 978-989-758-187-8

scientific language must denote things precisely. But to understand "information" were we offered a hierarchy of mystical fluids – data> information> knowledge>wisdom – each distilled from its predecessor in a chemical engineering metaphor. Even in the 1960s I was scornful of this idea, except as an imaginative point of departure[1]. Without a terminology with precise operational meanings, we cannot conjecture testable hypotheses from which to formulate theories for understanding and predicting the behaviour of systems employing that wonderful, new economic resource: information.

2 SIGNS AND SEMIOTICS

"Information" is a useless primitive concept; it has so many different meanings. Armed with the criterion "Take me to see some." I searched for a better primitive (Stamper, 1973). And there it stood: the sign. Semiotics (Nöth, 1990), the study of signs takes its name from the ancient Greek for a symptom (the sign of a disease), which must be something physical. From its roots in philosophy, semiotics has an extensive literature that few were bothered to read. John Locke (1690) had identified the "doctrine of signs" as the bridge between the physical and social worlds: our technical and organisational domains. Signs are things standing for other things that we want to communicate about. So, to displace DIKW's four mystical fluids, I wrote a book about information as a number of precisely defined properties of signs, all of them capable of empirical investigation. Three categories of them are well established in the literature[2], but I drew attention to two others[3] and added the social products of using signs to form a "semiotic framework" to divide an empirical science of organisational information into distinct areas of investigation. Incidentally, it serves as a checklist when working on any information system because, to be effective, it must function correctly on all six levels.

There is nothing mystical about signs. They always have a physical form, which may be investigated empirically in different ways, as indicated in this table. Technical properties do not depend on any human agent whereas the others always involve signs in relation to individuals or communities.

2.1 A Broader Focus

Can this broader understanding of information help us to improve upon the disgraceful track record for project failure? Every enterprise is coy about failures, so figures are very difficult to obtain, but trawling the web, as I last did in 2012, suggests, roughly speaking, that 25% succeed, 50% fail to meet functional requirements, budget or timing, while 25% are totally written off: a disgrace! Will a broad, unifying, scientific foundation help to eliminate or reduce those failures?

Each technical branch of semiotics has its own scientific support. Physics underpins hardware engineering; statistics and probability theory support work on the empirics of signs; while the formal sciences of logic and mathematics, as adapted by computer science, deal with the syntactic aspects of signs. Those excellent foundation disciplines tempt us to retreat into the safe hands of software engineering, well away from the messy domains of human and social behaviour. But the problems of engineering software for computers differ fundamentally from those of engineering information systems for organisations, unless you treat organisations as though they were computers with various information fluids flowing through them.

SEMIOTIC FRAMEWORK		**semantics of "information"**
Human Information Functions (neglected)		
	SOCIAL WORLD: norms, law, culture, attitudes, values, beliefs commitments,	***norms***
EFFECTIVENESS	***PRAGMATICS***: intentions, communications, conversations, negotiations	***sign-tokens***
	SEMANTICS: meaning, signification, denotation, connotation, validity, truth-falsehood	***sign-types***
	Technical Platform (dominant)	
	SYNTACTICS: formal structure, logic, language syntax, software, data, files	***sign-types***
EFFICIENCY	***EMPIRICS***: pattern, variety, noise, entropy, channel capacity, redundancy, efficiency codes	***populations of sign-tokens***
	PHYSICAL WORLD: signals, traces, hardware, speed, energy and material consumption, info economics	***sign-tokens***

A software engineer need not differentiate between a game about dungeons and dragons and a system affecting the lives or livelihoods of real people. Ensuring the safety of an atomic power station or providing social security for a population entail problems of meaning, intentionality and the social value of the signs. Only in relation with

[1] DIKW comes from TS Eliot's 1934 poem, *The Rock*. Science may start from imaginative ideas but must develop them with criticism and imaginative tools of other kind. However, a scientist who has access to poetic ideas gives me more confidence than one of constrained imagination.

[2] Eg: Syntactics, semantic and pragmatics in the writings of Charles Morris (1946) and CS Peirce (1931-35).

[3] CS Peirce included their physical properties and Colin Cherry (1957) their statistical properties.

those properties. Working on the analysis and design of an enterprise, with or without a computer application, one must deal rigorously with the real world (not formal) meanings of all the data, the intentions they express, and the agents who bear responsibility for their personal and social effects.

2.2 A Unifying Science

Organisational information systems engineering needs a unifying scientific discipline. To the technical branches of semiotics we must add appropriate treatments of semantics, pragmatics and the social properties of signs *but also with the essential precision and formality for our work*. Whereas the Taylor's 100 year-old tools serve the technical domains, they do not help us with meanings, intentions or the social properties of information, unless one counts adding informal comments to the documentation. The challenge is to clarify the essential human and social concepts and handle them in precise formal terms. Until we have without a rigorous science behind us, one that deals with organisations as well as computers, we shall continue to work on organisations as skilled artisans like the craftsmen who built early Rolls Royce cars, but unable to keep pace with change because organisations as they evolve to equate with Rolls-Royce aero-engines

2.3 Phases of Scientific Progression

How can we move forward? Thomas Kuhn (1970) has shown that science progress in two ways: in a Normal phase, while everyone works on a set of problems determined by a fixed paradigm with its dominant metaphor, taught from similar texts, until anomalies undermine the shared body of theory and a revolutionary phase is precipitated. Taylor's late 19th century techniques dominate our education and our practice but its anomalies are only beginning to disturb a few of us. Perhaps we imagined that fundamental changes were taking place while all we had were continuous, incremental adaptations of Taylor's methods and tools, via O&M of the interwar years, their adaptation for computer systems, followed by numerous modifications by software engineers that were unified in UML; but, beneath the surface, the old ideas remained in place.

Let us call to mind some of those anomalies, They include: an appalling project failure rate; persistence of sloppy ideas such as DIKW, inadequate treatment of meaning and intentionality, a weak understanding of how information delivers any value; high cost of system maintenance; obscure documentation that prevents an organisation's management from exercising control over projects; obscure mountains of documentation that make it difficult to involve an organisation's members from contributing to a system's design and development; a long lead time before a project can deliver benefits; and so on. Where is our scientific motivation?

If we had a serious scientific tradition and noticed that so much is wrong, we should be out on the proverbial streets in protest. Which makes me suspect that a lack of scientific spirit in the Information Systems community is holding back progress. Below I show that the comments of programme committee for another conference that expose their unawareness of scientific method and their responsibility to apply it.

My position is that it is time for a scientific revolution in our field. It is time for a new dominant metaphor and a better paradigm. Why doesn't everyone share my disquiet?

2.4 Resistance to Change

Perhaps Kuhn's explanation is enough: people who have expended decades acquiring expertise in some orthodox methods, for which they are hired at comfortable salaries, react against the threat of having to learn another way of working. Certainly, when consultancies build computer applications that need their expertise to maintain them, they benefit from a long-term, reliable cash flow; if all their competitors work within the same antiquated paradigm, their government and industrial clients have no alternative but to buy similar orthodox-style products from another consultancy. So why upset the boat? Those who teach the long-established orthodoxy react in a similar manner.

New ideas that threaten a comfortable way of life will nearly always come from a rather isolated maverick, so the opposition is easily attacked. When Max Planck's quantum theory encountered this treatment he said that science progresses one funeral at a time. We may feel great sympathy for him but should acknowledge the difficulty we all encounter when adopting a new paradigm.

So, having called for a revolution, I shall do something that you will probably consider even more foolish: I assert that there is a radically better paradigm for our work that can vastly improve our tragically bad project failure rate and it is based on a more suitable metaphor, one that embraces both the technical and the social aspects of the engineering problems we are required to solve.

3 CONJECTURE AND REFUTATION

Of course, I express myself this way to provoke you to attempt to falsify my risky assertion. Why? Because my research team and I adopted Karl Popper' scientific method of Refutationism: science progresses by bold imaginative leaps that formulate new universal hypotheses that must be expressed precisely enough to be capable of falsification by even a single particular empirical observation or experiment although no proof of a universal hypothesis will result from any number of particular empirical tests. In order that I may learn, I invite your criticism.

When the courses I established for the steel industry became the basis for the UK's national programme run by the British Computer Society and the National Computing Centre, I became an academic at the London School of Economics and my chance to apply a radically new paradigm had arrived.

3.1 A New Paradigm

Instead of the information flow paradigm, I adopted a different metaphor from physics: *field* instead of *flow*. It became rather obvious when examining the computerisation of the Department of Health and Social Security. I noticed that a single shelf for books could house all the Acts of Parliament and Statutory Instruments containing the legal norms defining what that huge organ of state must do. Only a minority of the legal norms governed routine bureaucracy and only some were worth automating. If we could express that small percentage in a suitable formalism, a computer might be able to interpret them, in effect turning the legal norms into the programs for supporting computer applications. The actual procedure was to translate the 1m shelf of legislation into library of 400 thick volumes of "clerical codes" that were then translated in orthodox flow specifications.

In addition to the legal norms, the people involved in the health and social security work also make use of the numerous social norms belonging to their shared background knowledge. So we recast our task: to define the knowledge people in this activity domain must share if they are to collaborate in an organised way.

Knowledge (note this precise definition) consists of social norms (culturally evolved informally as well as enacted as legal norms by Parliament) that express what things they deal with (perceptual norms), how that world functions (conceptual norms), how to judge things (evaluative norms) and how to act in different situations (behavioural norms). This knowledge field binds together the community involved into a system or institution that governs how they collaborate on the relevant, shared activity.

3.2 Refutable Hypotheses

That broad idea led to the evolution of

F: a formalism that can express any of the norms in question; and

P: a program to interpret the formalised norms

Conjecturing a version of F and its associated version of P, the research proceeded iteratively by pitting F and P against bodies of norms of increasing complexity, until they failed, as a result of which learned enough to make improved versions of F and P. The scientific investigation never ends because the latest hypotheses always invites attempts to refute them, but one may apply the formalism and interpreter as soon as they seem acceptable for an engineering task.

4 RESULTS

We have achieved more than we initially hoped for and we have been able to test the results on innumerable desktop case studies but only two substantial actual organisational applications. (From the point of view of the refutationist method, we should be attempting many such real applications but the opportunity to do so is not readily offered by businesses that, contrary to all the propaganda, are seldom entrepreneurial enough to take any risk.)

4.1 Two Business Applications

Case-I: University AdministrationIn one country, we built their administrative system (A) using our methods and tools for the first time and, over ten years, compared it with a corresponding system (B) in a different country in the same region. B employed modular software of orthodox design, perfected on 200+ similar applications worldwide. System-A was bespoke and did all and exactly what the organisation required; system-B, on the other hand, forced the organisation

- to change to suit the available software and/or
- to pay for additional expensive software modules and/or
- to have clerical staff process data in the margins of printouts.

Such solutions make adaptation to changing requirements even slower, and more costly. Over ten years, the comparative costs for System-A were [I hesitate to say this, lest I be disbelieved] 80% lower than for System-B. Adapting to changing requirements was quick, easy and cheap, turning a sclerotic organisation into an agile one. Moreover, because everyone found System-A easy to understand, experienced users with detailed knowledge could contribute to the design and on-going improvement of the system. Additionally, the sound theoretical foundations of our methods meant that many desirable feature were inbuilt whereas orthodox systems, must treat them as optional extras at additional cost: a full historical database; explicit semantic structures and associated error detection; specification of responsibilities; traceable records of all error treatment; multi-lingual facility (English and one other language but any number of others could be added easily).

Case II: A Complex Expert System - This system was being developed using the best of orthodox methods but was on the point of being totally written off because the experts commissioning it could not understand what was being constructed for them. The orthodox documentation had grown to its usual gargantuan volume; with its impenetrable style, the experts could not understand much of it; it was boring to read and difficult to verify. So they invited two members of our team to apply our methods.

The documentation shrank to about one-twentieth of its original volume. The expert commissioners found the new formalism succinct and easy to read. They could see what the system designers were proposing and were able to steer the emerging system toward their goals. Implementation went through smoothly and successfully,

4.2 Criteria of Progress

You may not think that I have described anything resembling a revolution in our scientific field. That is exactly the right attitude. Refutationism demands permanent scepticism on the part of its practitioners. Despite that, Popper advises one to conjecture "bold hypotheses" that shift one's perspective in a surprising way. Better still they should preferably:

- explain as much as the hypothesis it is intended to replace;
- do so more succinctly,
- replacing a large obscure model with one that is simpler and easier to understand; and
- in a way that explains more about the domain;
- preferably bringing to light new invariants in the domain; while
- raising new, exciting lines of enquiry and application

4.3 Success? or Not yet?

The question: does the "knowledge field" paradigm achieve all that?

Given an organisation specified as a knowledge field, any number of suitable information flows can be derived from it but not the reverse.

Case II achieved a massive reduction in the documentation while making it easier to understand, thus reversing the plan to write off the project;

- a flow model tells you a lot about boring bureaucratic activity whereas the field model tells one what should happen for business reasons, especially who is responsible and mostly why; it contains a semantic model, it accounts for human intentions and, by showing the intended changes of attitudes, deals with the valuable products of the information;
- the semantics for the domain are contained in a Semantic Normal Form that is largely invariant over time and between cultures; the classification of norms enables one find a stable organisational kernel that remains invariant over all bureaucratic revisions that do not change the essential business activities;
- the computer-interpretable specification opens up a range of organisational research opportunities and practical products such as a touchstone to test any new computer application; with a Parliamentary Counsellor we have tested the method for legal drafting and parallel design of supporting software; it leads to ERP solutions based on 'atomic' modules; etc. etc.

5 SCIENTIFIC CRITICISM WELCOME

In conclusion, I present my position to you and explicitly ask for your critical questioning. In the best scientific tradition, I want you to take my request seriously and make your comments rationally and, therefore, capable of rational response. Recently, from another conference, the reviewers of my paper made unhelpful comments that were:

- value judgements to which no rational response was possible; or

- assertions that a statement or explanation is wrong or questionable without even a hint of why; or
- complaints that I did not cite their favourite authors who, in fact, we had read but found irrelevant to our work; or
- complaints about missing explanations that were actually in the text; while others
- complained that I had relied on their appropriate prior knowledge to keep my explanations to a length appropriate to a book rather than a conference-length paper;

This made me sceptical about our community having a well-established scientific tradition. If one, as a scientist (PC member, for example) writes a criticism of a scientific document, then one has a duty to abide by the same standards of discourse we impose on the authors.

Now is the time for some refutations! I hope I have provoked you into having interesting discussions. It would be unwise of colleagues younger than me to be so controvertial but I have reached a point in life when worrying about my future career would be pointless. Have fun!

ACKNOWLEDGEMENTS

Many members of the research team since 1971 deserve acknowledgement but I only have space to mention: Kecheng Liu and Yasser Ades who were responsible for the two major practical case-studies.

REFERENCES

Cherry, Colin, 1957, *On Human Communication,* Cambridge Mass, MIT Press

Kuhn, Thomas S., 1962, 1970, *The Structure of Scientific Revolutions*, Chicago, Chicago University Press.

Locke, John, 1690/1959, *Essay Concerning Human Understanding*, unabridged edtion 1959, Dover, New York.

Morris C., l946, *Signs, Language and Behaviour*, New York, Prentice Hall - Braziller.

Nöth, W. 1990, *Handbook of Semiotics*, Bloomington, Indian University Press

Pierce C. S., l93l-35, *Collected Papers*, (6 volumes), Hartshorne C. & P. Weiss (eds.), Cambridge, Mass. Harvard U.P.

Popper, Sir Karl, 1934/1959, *The Logic of Scientific Discovery*, London, Hutchinson.

Popper, Sir Karl, 1963, *Conjectures and Refutations*, London, Routledge and Kegan Paul

Stamper, R. 1973 *Information in Business and Administrative Systems*, Batsford, London & Wiley, New York.

Stamper, R, 2012 "A New Framework for IS Thinking and a Game for Teaching Organisational IS Rather than Business Applications of IT," Proc. UKAIS, New College, Oxford

Decision Criteria for the Payment of Technical Debt in Software Projects: A Systematic Mapping Study

Leilane Ferreira Ribeiro[1,2], Mário André de F. Farias[3,4], Manoel Mendonça[4] and Rodrigo Oliveira Spínola[1,5]
[1]*Graduate Program in Systems and Computer, Salvador University, Salvador, Bahia, Brazil*
[2]*Federal Institute of Bahia - IFBA, Jequié, Bahia, Brazil*
[3]*Federal Institute of Sergipe, Lagarto, Sergipe, Brazil*
[4]*Federal University of Bahia, Salvador, Bahia, Brazil*
[5]*Fraunhofer Project Center for Software and Systems Engineering at Federal University of Bahia, Salvador, Bahia, Brazil*
leilaneferreira@ifba.edu.br, mario.andre@ifs.edu.br, {manoel.g.mendonca, rodrigoospinola}@gmail.com

Keywords: Technical Debt, Technical Debt Management, Decision-making Criteria, Software Maintenance, Systematic Mapping.

Abstract: The term Technical Debt (TD) is used to describe the debt that a development team incurs when it takes shortcuts in the software development process, but that may increase the complexity and maintenance cost in the long-term. If a development team does not manage TD, this debt can cause significant long-term problems such as high maintenance costs. An important goal of the management of the debt is to evaluate the appropriate time to pay a TD item and to effectively apply decision-making criteria to balance the short-term benefits against long-term costs. However, although there are different studies that have proposed strategies for the management of TD, decision criteria are often discussed in the background and, sometimes, they are not even mentioned. Thus, the purpose of this work is to identify, by performing a systematic mapping study of the literature, decision-making criteria that have been proposed to support the management of TD. We identified 14 decision-making criteria that can be used by development teams to prioritize the payment of TD items and a list of types of debt related to the criteria. In addition, the results show possible gaps where further research may be performed.

1 INTRODUCTION

The term Technical Debt (TD) is used to describe the debt that a development team incurs when it takes shortcuts in the software development process, but that may increase the complexity and maintenance cost in the long-term (Brown *et al.*, 2010) (Kruchten *et al.*, 2012). In this work, we use the term "TD item" to refer to an instance of TD.

According to Brown *et al.* (2010), if a development team does not manage a TD item, this debt can cause significant long-term problems such as high maintenance costs. In this sense, effective management of TD is an important step to achieve a good quality in the software maintenance (Guo *et al.*, 2014).

Management strategies have been proposed in order to minimize negative impacts of management of debt. The main goal of these strategies is to evaluate the appropriate time to pay a TD item, i.e. the time for the development team change the system and eliminate the debt. Thus, knowing decision criteria used to choose the most suitable time for the payment of TD items is important to balance their short-term benefits against long-term costs.

Although there are different studies that have proposed strategies for the management of TD (Snipes *et al.*, 2012) (Seaman *et al.*, 2012) (Power, 2013) (Codabux and Williams, 2013) (Guo *et al.*, 2014) (Mamun *et al.*, 2014), none of them provides a deep discussion on decision-making criteria for the payment of TD. On these works, decision criteria are often discussed in the background, sometimes they are not even mentioned. Thus, despite their importance, there is not a comprehensive view on the existing criteria.

In this context, this paper presents a systematic mapping study over studies published up to 2014 that focus on management strategies of TD. This allowed us to investigate how researches are being conducted in this field and to address the following research questions:

Ribeiro, L., Farias, M., Mendonça, M. and Spínola, R.
Decision Criteria for the Payment of Technical Debt in Software Projects: A Systematic Mapping Study.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 572-579
ISBN: 978-989-758-187-8

- ***RQ1.*** *What decision-making criteria have been proposed for the payment of TD?*
- ***RQ2.*** *What are the types of TD related to the decision-making criteria for the payment of TD?*
- ***RQ3.*** *Which empirical evaluations have been performed to evaluate the criteria?*

We held searches in three digital libraries (ACM Digital Library, IEEE Xplorer, and Scopus). 38 studies were considered relevant to answer the research questions. The results provide a list of 14 criteria that can be used to support decision-making on the payment of TD, and a list of types of TD that have been considered in approaches that focus on the payment of debt.

Besides this introduction, this paper has six other sections. Section 2 discusses some related work. Section 3 details the systematic mapping method. Next, in section 4, the results of the mapping study are presented. Some implications of this work for researchers and practitioners are discussed in section 5. Next, Section 6 shows the threats to validity. Finally, Section 7 presents the conclusions and directions for future researches.

2 RELATED WORK

In this section, we present other secondary studies in the TD area.

The study performed by Tom *et al.* (2013) reported an exploratory case study that involves multivocal literature review, supplemented by interviews with software practitioners and academics to consolidate understanding of the nature of TD and its implications for the software development. The results of this study included the creation of a useful theoretical framework, consisting of a set of TD dimensions, attributes, precedents and outcomes, as well as the phenomenon itself and a taxonomy that describes and encompasses different forms of TD.

Villar and Matalonga (2013) performed a systematic mapping study in order to understand the feasibility of using the TD metaphor as a tool for project management. The main purpose was to identify the current state of TD definitions. The results show that there is no agreed definition of the technical debt term.

In another systematic review, Ampatzoglou *et al.* (2015) investigated how the financial aspects are defined in the context of TD and how they are related to the concepts of software engineering. The results indicate: (i) the most common financial terms used in TD researches: principal and interest, and (ii) the financial approaches that have been more frequently applied for managing TD: real options, portfolio management, cost-benefit analysis, and value-based analysis. Furthermore, the authors emphasize that the application of such approaches lacks consistency, i.e., the same strategy is differently applied in different studies, and in some cases lacks a clear mapping between financial and software engineering concepts.

In another work in this area, Li *et al.* (2015) conducted a systematic mapping in order to obtain a comprehensive understanding of TD and an overview of the current state of research on its management. The results pointed out 10 types of TD, 8 TD management activities, and 29 tools for TD management.

In this same sense, Alves *et al.* (2016) performed a systematic mapping study. Their results include an initial taxonomy of types of TD, a list of indicators that was proposed to identify TD, management strategies, and an analysis of the current state of the art, which allows to identify possible gaps and research topics.

These studies are different from the mapping study presented in this paper. They provide a broad view of the TD management through different perspectives. This work focuses on identifying a set of criteria to be used in the decision-making on the payment of TD items. Therefore, our mapping study and the works discussed above are complementary to each other.

3 SYSTEMATIC MAPPING METHOD

Systematic mappings are used to evaluate and interpret relevant works relating to a research question, an area or a phenomenon of interest (Kitchenham and Charters, 2007). A systematic mapping study follows a set of well-defined steps, according to a protocol, to reduce the bias inherent in an informal review of the literature (Petersen *et al*., 2008). We chose to conduct a mapping study because it allows accessing and analyzing the primary studies aiming to summarize the evidences related to our research questions and carry out future researches. We describe the steps of the mapping method below.

3.1 Research Questions

Our general purpose is to better understand the decision-making criteria on the payment of TD

through a systematic mapping study. Thus, we defined three research questions which guide this study and reflect our goals. These questions and their motivations are described at the following:

RQ1. *What decision-making criteria have been proposed for the payment of TD?*

In order to achieve the software quality, a TD item must be effectively managed. In this sense, evaluating whether a TD must be paid and the suitable time for this may reduce the negative impacts of debt on the quality of the software project. Knowing decision criteria used to choose the appropriate time to pay off the debt may support this task.

This question intends to identify and classify these decision criteria.

RQ2. *What are the types of TD related to the decision-making criteria for the payment of TD?*

A TD item can be inserted at any moment in the software development life cycle and may be related to several immature artifacts such as bad design, incomplete documentation, and missing tests. These immature artifacts may be seen as a type of debt that may burden software maintenance in the future (Alves *et al.*, 2016).

Different types of debt can bring different consequences to the software project, influencing what we need to consider when deciding if a debt should be paid and when.

In order to effectively manage TD, it is important to know the relation between types of debt and decision criteria. Thus, the purpose of this question is to identify types of TD that have been studied in the works that focus on debt payment criteria.

RQ3. *Which empirical evaluations have been performed to evaluate the decision criteria?*

Alves *et al.* (2016) reported that most of the proposals in the TD management area still require more empirical evaluation. In this context, this question investigates which types of validation have been used in studies that focus on decision criteria for the payment of TD. This information is important to analyze the level of the maturity of the proposed approaches.

3.2 Search Strategy

In consonance with Petersen *et al.* (2008), the first step in conducting the mapping study is to look for primary studies into the defined scope. To define the search string, we considered the following aspects and keywords:

- Population: Technical Debt;
- Intervention: management of TD;
- Results: methods, criteria, and process to support decision on payment of TD.

We used these keywords and OR and AND operators to assemble the terms. Table 1 presents the complete search string used in this work. We applied the search string to titles and abstracts in some digital database. We did not use full text search because full text search resulted in a very large number of studies from domains other than software projects. The search covered papers published up to 2014.

3.3 Databases and Study Selection

We chose three digital libraries to the search process: (i) ACM Digital Library, (ii) IEEE Xplore, and (iii) Scopus. We selected these databases because, according to Alves *et al*. (2016), they have a large concentration of studies in the TD area.

To support the study selection process, we defined the following inclusion and exclusion criteria:

- *Inclusion Criteria*: the study needs to explore a theory, a practice, or an approach related to the management of TD.
- *Exclusion Criteria*: we excluded studies that do not address management of TD. Surveys and secondary empirical studies were removed, since they report approaches from others. Challenges, showcases, and abstracts were also excluded, such as Tamburri *et al.* (2013) and Shah *et al.* (2014).

The selection of papers was divided into three steps. Figure 1 shows the selection process. After the search, we had 450 studies, published between 1991 and 2014. In the first step, we removed the duplicate studies. Next, we read the titles and abstract of resulting selection in order to analyze if the papers were into our scope. Finally, in the last step, we completely read each study in order to analyze it.

The first step returned 332 studies. The second step reduced the list to 61 papers. Our final step resulted in 38 studies, published between 2010 and 2014, to be further analyzed and classified. A whole list of the studies is available at https://goo.gl/RivQ16. Table 2 shows the number of papers by publishing type.

Table 1: Search String.

Population	(("Technical Debt")
	AND
Intervention	(Management OR Monitoring OR Control)
	AND
Results	(Criteria OR Method OR Process))

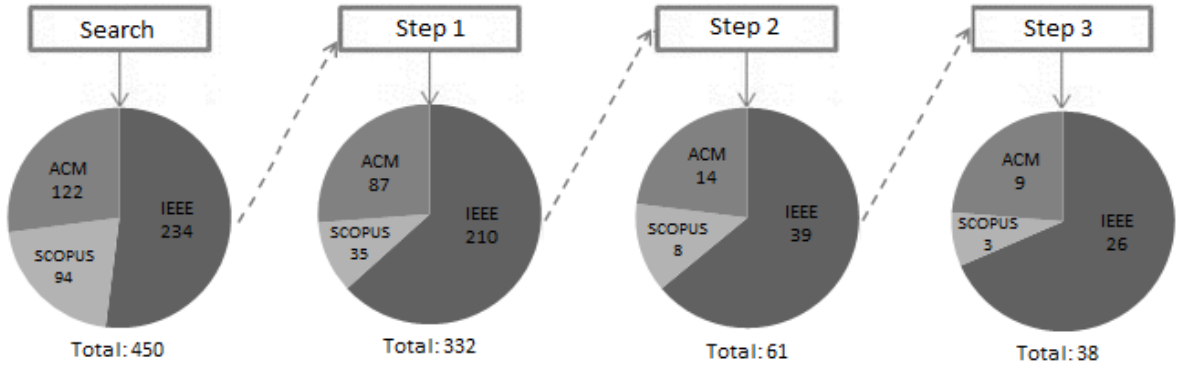


Figure 1: Study selection process.

3.4 Classification Scheme

We defined three categories to classify the papers and answer the research questions:

- **Decision-making Criteria on Payment of TD (RQ1):** in order to classify the criteria, a researcher collected the decision-making criteria and their definitions following the terminology straight from the studies. We assumed as a criterion the strategy that supports decisions about when and if a TD item should be paid;
- **Types of TD Related to the Decision-making Criteria (RQ2):** this category lists the types of TD that were related to any criterion in the studies. We used the types of debt proposed by Alves *et al.* (2014);
- **Empirical Evaluation (RQ3):** we verified whether the proposed criterion has been evaluated through empirical methods and, if so, which method was used. We considered that a study has an empirical evaluation if it brings at least one section with some discussion dedicated to this topic.

Table 2: Number of papers by publishing type.

Type	Number of papers
Conference	18
Journal	9
Symposium	1
Workshop	1

4 RESULTS

This section presents the main results of the data extraction activity. The extracted data were recorded on a spreadsheet that is available at https://goo.gl/Akdu8r. We analyzed the extracted data in an effort to answer our research question.

4.1 Decision-making Criteria (RQ1)

In this section, the decision-making criteria found in the literature and their definitions will be presented. We classified them into four categories:

- **Nature of the TD:** criteria that are related to the TD's properties, such as their severity and time when the debt was incurred;
- **Customer:** criteria into this category concern about the impact that debts have on the customers;
- **Effort:** criteria that are related to the cost of TD, such as the impact of the TD on the project and what effort will be applied to pay the TD item.
- **Project:** criteria that are related to the projects' properties, such as their lifetime and their possibility of evolution.

These categories may help the development team on better understand the decision criteria and decide the suitable time to pay off a TD item. For instance, in a specific situation, it may be more important for the team to prioritize the customer category. Thus, criteria related to the category customer may be applied in order to perform the management of TD items. On the other hand, whether the cost to pay a TD item is more important than its impact on costumer, criteria related to category effort will be more relevant to decide which and when a TD will be paid.

We identified 14 decision-making criteria to support the choice of the suitable time for the payment of debt. Table 3 presents criteria found in this mapping study (sorted by category), as well as their definitions, and the papers that discussed each one.

Figure 2 shows criteria distribution over the investigated years. From this figure, we highlight two outcomes:

(i) *Debt impact on the project* and *Cost-Benefit* are the most explored criteria by the analyzed studies (both studies had 8 citations). Moreover, they appear nearly every years covered by this mapping. This may indicate that the biggest concern at the moment of decision-making on payment of a TD item is the impact and extra cost that a debt may cause on the project;

(ii) most criteria have clearly been not much explored. Five criteria were approached by two studies and other four only by one study. In this same sense, decision-making criteria were covered in less than 50% (17 from 38 papers) of the studies that focused on management of TD. This set of results indicates that these criteria need further investigation in order to improve their maturity.

4.2 Types of TD Related to Decision-making Criteria (RQ2)

In order to answer this question, we identified types of TD that were discussed with regards to the decision-making criteria. Table 4 presents the relation between types of TD and criteria. We can see that although many types of TD had already been discussed in several researches, only Defect Debt and Design Debt were related to criteria. As different types of debt can bring different consequences to the software project, influencing what we need to consider when deciding if a debt should be paid and when, the lack of relation between other types of debt and decision criteria provides us the following open question: "*Are criteria independent of types of TD or there is some kind of influence between them?"*. We do not have evidences to answer this question. This gap needs to be explored by academics in further researches.

Table 3: Decision-making criteria.

Category	Criteria	Definition	Studies
Nature of the TD	Severity of the Debt	Debt items with high level of severity should be paid.	S1, S9
	Existence of workaround	The payment of debt items that have a workaround may be delayed.	S1, S14, S9
	Existence time of debt items in the project.	Debt items that are a long time in the project should be paid.	S9
	Localization of TD	If the debt is located in a resource that will change due to a development or maintenance activity, the software engineer should take advantage of the change to pay the debt.	S38
Customer	Visibility	The visible debt must be paid.	S5
	Analysis when the refactored part will be used	Pay debt items that are in widely used parts of the system.	S4, S33
	Debt impact on customer	Debt items that impact directly on the customer should be prioritized.	S1, S9
Effort	Debt impact on the project	Debt items that offer the greatest impact on the project should be paid.	S2, S3, S5, S8, S9, S1, S24, S38
	Scope of tests	Debt items with smaller scope of tests to validate their adjustment should be prioritized.	S1, S9
	Cost-Benefit	Debt items with good cost-benefit should be paid. If the cost of the debt is less than the cost of paying it off, the payment can delayed.	S1, S14, S4, S10, S23, S9, S24,S28
	Effort to implement the proposed correction	Debt items that require less effort to be paid must be removed first.	S1, S3, S9
Project	Nature of the project	Debt items of critical projects must be paid quickly.	S11, S24
	Lifetime of the system	Debt items in projects that will be discontinued soon should not be paid.	S12, S14, S24
	Need of evolution of the system or features	Debt items of systems or modules that will stop evolving or is stable and will not be affected by future changes should not be paid.	S21

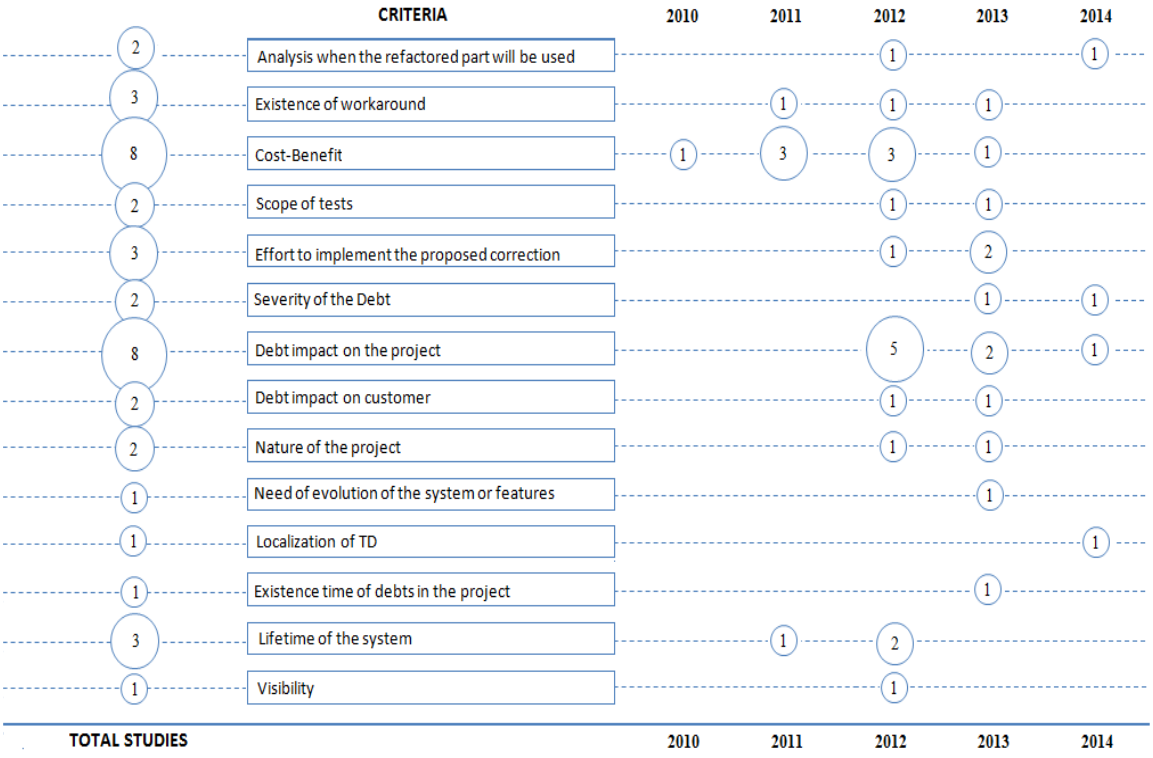


Figure 2: Criteria distribution over the investigated years.

4.3 Empirical Evaluations (RQ3)

We analyzed whether the papers investigated in this mapping study have conducted some type of empirical evaluation to validate the proposed criteria. Despite we identified different criteria in 17 papers, none of them has been evaluated through an empirical study.

According to Novais *et al.* (2013), empirical evaluation of technologies has increased significantly in the software engineering domain over the last years. However, we cannot observe this regarding studies that focus on decision-making criteria for the payment of TD. This implies that proposed criteria still require empirical investigation, so that their benefits and limitations can be known with increased confidence.

5 DISCUSSION

5.1 Implications for Practitioners and Researchers

The results of this mapping study point to the following implications for practitioners:

- We identified 14 decision criteria that can be used to decide and/or prioritize the payment of TD items incurred in software projects. After identifying TD items, developers can apply the criteria to each of them and decide on the payment of that item;
- We defined 4 categories to facilitate the understanding and using of the criteria: nature of TD, customer, effort, and project. Software engineers can use these categories in the initial phases of a strategy for managing the TD in their projects.

For researchers, the findings of this mapping study point to the following implications:

- Different criteria were mapped, however, we did not identified any empirical study to assess them. This indicates that the criteria still require evaluation, so that their benefits and limitations could be known;
- Although there are many types of TD, only two of them have been discussed with respect to decision-making criteria. Thus, the results were not conclusive as regards to the relation between decision criterion and types of TD. This gap needs to be further investigated.

Table 4: Relation between types of technical debt and decision criteria.

Types of TD	Criteria	Studies
Defect Debt	- Severity of the Debt - Existence of workaround - Debt impact on customer - Debt impact on the project - Scope of tests - Cost-Benefit -Effort to implement the proposed correction	S1, S16, S6
Design Debt	- Debt impact on the project - Analysis when the refactored part will be used - Cost-Benefit	S2, S4, S25

6 THREATS TO VALIDITY

Our study has some threats to validity. We present them below with the strategies for its mitigation.

Selection Bias: we selected each study based on the judgment of the inclusion and exclusion criteria. Thus, we cannot guarantee that all relevant primary studies were selected. With the intention of mitigate this threat, we discussed the study protocol among the researchers to guarantee a common understanding and searched the studies into the main digital libraries in our field.

Data Extraction: bias or problems on data extraction from selected studies can affect their classification. In order to reduce this bias, we discussed deeply the definitions of data items and the classification scheme.

External Validity: we carried out a systematic mapping study over studies published up 2014 that focused on TD management. This implies that we might have missed some relevant studies. Thus, we cannot generalize our conclusions for whole TD management approaches. However, our outcomes allow us to draw insights to guide further investigations.

7 CONCLUSIONS

The goal of this work was to conduct a systematic mapping study of the literature in order to identify criteria to support the decision on the payment of existent TD items in software systems. We focused on studies published up 2014 and selected 38 primary works that discuss TD management strategies.

The main contribution of this work was the identification of 14 decision criteria that can be used by development team to decide and/or prioritize the payment of TD items. In addition, we identified that only two types of TD were related to decision-making criteria. In this sense, we cannot recognize whether: (i) decision criteria are independent of types of TD, or (ii) there is some kind of influence between decision criteria and types of TD.

Considering evaluation methods, we identified that none of analyzed studies has performed any kind of empirical evaluation. This may indicate a low level of maturity of the decision-making criteria for payment of TD.

In general, the results provide some evidence and motivation for continuing to study decision criteria for TD payment. As future work, we will investigate the gaps identified in this mapping study. In particular, continuing to explore decision criteria in order to answer the following question: *Are criteria independent of types of TD or there is some kind of influence between them?*. We also intend to work on the development of a TD management strategy based on the identified criteria and their combinations.

ACKNOWLEDGEMENTS

This work was partially supported by CNPq Universal 2014 grant 458261/2014-9.

REFERENCES

Alves, N.S.R., Ribeiro, L.F., Caires, V., Mendes, T.S. & Spínola, R.O., 2014. Towards an Ontology of Terms on Technical Debt, In the *Sixth International Workshop on Managing Technical Debt,* Victoria, British Columbia.

Alves, N. S., Mendes, T. S., de Mendonça, M. G., Spínola, R. O., Shull, F., & Seaman, C, 2016. Identification and management of technical debt: A systematic mapping study. *Information and Software Technology, 70*, 100-121.

Ampatzoglou, A., Ampatzoglou, A., Chatzigeorgiou, A., Avgeriou, P. 2015. The financial aspect of managing technical debt: *A systematic literature review, Information and Software Technology, Volume 64*, Pages 52-73, ISSN 0950-5849.

Brown, N., Cai, Y., Guo, Y., Kazman, R., Kim, M., Kruchten, P., Lim, E., MacCormack, A., Nord, R., Ozkaya, I., Sangwan, R., Seaman, C., Sullivan, K. & Zazworka, N., 2010. Managing Technical Debt in software-reliant Systems, a, Proceedings of the *18th FSE/SDP Workshop on Future of Software Engineering Research*, 47-5.

Codabux, Z. & Williams, B., 2013. Managing technical debt: An industrial case study. In: *4th International Workshop on Managing Technical Debt (MTD)*.

Guo, Y., Spínola, R. O. and Seaman, C., 2014 . Exploring the costs of technical debt management – a case study on *Empirical Software Engineering, v. 1*, p. 1-24.

Kitchenham, B. A. & Charters, S. 2007. Guidelines for performing systematic literature reviews in software engineering. *Tech. Rep. EBSE-2007-01,* KeeleUniversity.

Kruchten, P., Nord, R. L., Ozkaya, I., 2012. Technical Debt: From Metaphor to Theory and Practice. *IEEE Software, 29(06)*, 18-21.

Li, Z., Avgeriou, P. & Liang, P., 2015. A systematic mapping study on technical debt and its management. In *Journal of Systems and Software, Volume 101*, Pages 193–220.

Mamun, M. A., Berger, C. & Hansson, J., 2014. Explicating, Understanding and Managing Technical Debt from Self-Driving Miniature Car Projects, In: *30th IEEE International Conference on Software Maintenance and Evolution (ICSME)*.

Novais, R.L. et al. 2013. Software evolution visualization: A systematic mapping study. Information and Software Technology. 55, 11 (Nov. 2013), 1860–1883.

Petersen, K., Feldt, R., Mujtaba, S. & Mattson, M., 2008. Systematic mapping studies in software engineering, In the *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering*, University of Bari, Italy.

Power, K., 2013. Understanding the impact of technical debt on the capacity and velocity of teams and organizations: Viewing team and organization capacity as a portfolio of real options. In: *Managing Technical Debt (MTD)*.

Seaman, C., Guo, Y., Zazworka, N., Shull, F., Izurieta, C., Cai, Y. & Vetro, A., 2012. Using technical debt data in decision making: Potential decision approaches, *Third International Workshop on Managing Technical Debt (MTD)*.

Shah, S. M. A., Torchiano, M., Vetrò, A. & Morisio, M., 2014. *Exploratory testing as a source of technical debt, IT Professional, vol. 16, no. 3,* Article ID 6475929, pp. 44-51.

Snipes, W., Robinson, B., Guo, Y. & Seaman, C., 2012. Defining the Decision Factors for Managing Defects: A Technical Debt Perspective. In: *3th International Workshop on Managing Technical Debt (MTD)*.

Tamburri, D.A., Kruchten, P., Lago, P. & Van Vliet, H., 2013. What is social debt in software engineering?, In: *6th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)*, pp. 93-96.

Tom, E., Aurum, A. & Vidgen, R. B., 2013. An exploration of technical debt, *Journal of Systems and Software 86(6)*, 1498-1516.

Villar, A. & Matalonga, S., 2013. Definiciones y tendencia de deuda técnica: Un mapeo sistemático de la literatura. Anais do CIBSE13 - Congresso Ibero-Americano em Engenharia de Software, Montevideo, Uruguai, Abril 8, 9 e 10, pp 33-46.

Decision Criteria for Software Component Sourcing
Steps towards a Framework

Rob J. Kusters, Lieven Pouwelse, Harry Martin and Jos Trienekens
Faculty MST, Open University, Valkenburgerweg 177, Heerlen, The Netherlands
{rob.kusters, harry.martin, jos.trienekens}@ou.nl, lpouwelse@gmail.com

Keywords: Make-or-buy, Software Components, Open Source, Closed Source, in House Development.

Abstract: Software developing organizations nowadays have a wide choice when it comes to sourcing software components. This choice ranges from developing or adapting in-house developed components via buying closed source components to utilizing open source components. This study seeks to determine criteria that software developers can use to make this choice. Answering this question will result in a list of criteria that can, after further validation, be used to develop structured decision support in this type of decision. A first step is a literature search resulting in an initial list. Since the literature used was not specifically targeted at the question at hand, it was decided to separately conduct interviews to obtain an independently derived list of criteria. In a second part of the interview the respondents were confronted with the list resulting from literature. Together this resulted in a preliminary proposal for decision criteria for software sourcing.

1 INTRODUCTION

Delivery in time and within budget of (business) software that meets the functional and quality requirements is often a challenge. Component-based software development is often used to deal with this challenge but the selection of appropriate software components then becomes an important decision (Jha et al., 2014). A component can be defined as a coherent package of software that can be independently developed and delivered as a unit, and that offers interfaces by which it can be connected, unchanged, with other components to compose a larger system (D'Souza and Wills, 1997). When developing component-based software, an organization nowadays has a wide choice of sourcing options. The main choices are:

- In-house development
- Re-use (possibly with adaption) of earlier in-house developed components
- Acquisition of commercial components
- Usage of open source components
- Adaption of open source components.

Choosing between these options is not obvious (Cortellessa et al., 2008). However, we were unable to find a good overview of criteria that could be used for such a decision. In this paper we propose a first attempt at filling this gap. A two-fold approach is taken. First, in a literature survey we try to identify a basic list of criteria. After this we conducted a series of interviews with experienced software developers and managers without using the results obtained from literature. From this, a list of criteria derived from practice is extracted. In a second part of the interviews, the results from literature are discussed explicitly. We expect that the results can be used as the basis for the development of structured decision process support for sourcing software components.

In section 2 related work is discussed. The methodology used in the research is described in section 3, and execution of the research and the results in section 4. The paper ends with conclusions and a discussion of results in section 5.

2 RELATED WORK

A significant body of literature is already available on management of software development in general. However most of the literature found is only indirectly related to software component sourcing. No specific literature on software component sourcing decision criteria was found. We did find however literature on relevant aspects, focusing at the basic make-or-buy decision, at the consequences of organizing re-use of in-house developed components, or on the advantages and disadvantages of using open

Kusters, R., Pouwelse, L., Martin, H. and Trienekens, J.
Decision Criteria for Software Component Sourcing - Steps towards a Framework.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 580-587
ISBN: 978-989-758-187-8

source which will be a topic of discussion in this article as well.

Morisio et al., (2002) shows the main issues of a 'make or buy' decision. They also state that each product variant has its specific considerations regarding appropriate requirements, risks and costs. Cortellessa et al., present a framework supporting the choice between selection of commercial component software and development in-house (Cortellessa et al., 2008). Daneshgar et al have examined, in relation to the 'make or buy' decision on the basis of 10 existing decision criteria, additional criteria that affect small and medium businesses (Daneshgar et al., 2013). Boehm and Bhuta (2008) also identified advantages and disadvantages of commercial off-the-shelf products in their study. The choice between the use of existing software components rather than to fully develop in-house is often made implicitly. Also, in projects that have been studied, it is implicitly expected that the development time and effort required can be reduced by making use of software components. However, convincing evidence is not yet available (Morisio et al., 2002). They also note that when using commercial off the shelf products new types of activities and their associated costs have to be taken into account (Morisio et al., 2002). The suitability of a component-based software system is highly dependent on the architecture of the system. Consistency and coupling play an important role in determining the quality of the system in terms of reliability and the effect of the component on the maintainability and availability of the system as a whole. This is also an area that should be taken into account when making sourcing decisions (Jha et al., 2014).

From the point of view of re-use as a sourcing option, re-use has the potential to shorten lead times, improve quality and reduce development costs. The studies conducted by (Lim, 1994) and (Kakarontzas et al., 2013) suggest that reuse of software results in higher quality. Also (Favaro et al., 1998) indicate that the economic benefits of software reuse are substantial. However, the reuse of software has shown to be challenging for many organizations on both a technical and organizational level (Kakarontzas et al., 2013). Results from the study of (Lim, 1994) are broadly consistent with research from (Kakarontzas et al., 2013). Lim mentions the following arguments for reuse: reduced delivery times; lower development costs and higher quality by fixing bugs in the product, but balances this with the need for sufficient funding for developing, maintaining and keeping components for re-use available. Frakes (2005) further elaborates the organizational issues that need to be dealt with for facilitating reuse.

Also, work is available focusing on the adoption and the adaption of open source components. Such components have numerous benefits including free customizable source code. On the other hand, the use of (open source) software components may present various challenges concerning selection, testing and integration. If a system is being distributed or sold, it is e.g. important that a component with an appropriate license is selected (Chen et al., 2007). Ruffin and Ebert (2004) also state that, depending on the product, use, and market conditions, certain open source properties may be advantageous. One example is the existence of a large user community which results in a de-facto standard. Additionally, like (Chen et al., 2007) they emphasize the importance of adhering to license conditions. There seems to be general disagreement on the added quality open source can provide. The study by (Paulson et al., 2004) suggests that defects are generally found and resolved faster in open source than in closed source software. Ruffin and Ebert (2004) argue that open source software may increase security. However, (Schryen and Kadura, 2009) state that this conclusion requires further research, since a solid basis for this conclusion has yet to be established. For users of commercial off the shelf software components, it is more difficult to track changes than for open source software users. Open source users are also more concerned about the reputation of their support provider (Li et al., 2006).

In the related field of package software selection a good overview is provided by (Jadhav and Sonar, 2011) which gives an interesting and very possibly relevant overview of package selection criteria. However, the field is sufficiently distinct to prevent us in this stage from accepting these results as-is. They can however be looked into in a further stage of the research.

All together we see significant contributions, often focusing on specific but related aspects of the sourcing criteria issue without, as of yet, resulting in a well-structured overview of criteria. Based on this conclusion, we decided to investigate sourcing criteria in a dedicated study.

3 METHODOLOGY

We started with a literature search using relevant combinations of the search terms “advantages, disadvantages, open source software, closed source software, software components, software component

selection, software reuse, and software 'make or buy'". We used the generic search engine scholar.google.com and also the following online databases:

- ACM Digital Library
- IEEE Digital Library
- JSTOR Business, Biological, Mathematics & Statistics Collection
- ScienceDirect (Elsevier)
- SpringerLink

Relevancy of papers was assessed first on title and abstract. Resulting interesting papers were then investigated in detail. Selected papers were also used as the basis for a further reference based search (forward and backward) to find additional related papers. In total 94 papers were selected of which after further study 11 were eventually used.

The results of the literature search were not convincing. Meaning that many criteria had to be derived from a literature base that was not specifically written for our purpose. So although the first resulting list of criteria might look plausible, we felt it had insufficient justification to serve as the sole basis for this research. Straightforward validation of such a list (e.g. in a survey) could provide information on the relevance of items already on the list, but it would be unlikely to lead to adding missing items to this list. To substantiate our first impression of the literature search and to gain insight into the way in which these and local criteria are being interpreted in practice, we opted for an in-depth case study.

Since both relevance and completeness are relevant objectives for such a type of research, we decided on a twofold approach. In an interview, first an open part took place aimed at independently identifying a set of criteria that can provide a reference set for discussion and valuation of the set derived from literature. This was followed by a second, semi-structured part. Here, explicitly based on the list resulting from literature, we made a first attempt to assess the potential relevance of the literature set.

The choice was made for open in-depth interviews since we felt the questions were too complex to allow sufficient quality results and to provoke sufficient response from a survey. We felt the disadvantage of limited participation was off-set by the depth and quality of the results which we could expect from in-depth interviews.

We looked at a single organization where component based development had been in use for several years and where the sourcing decision is therefor made routinely and where alternate sourcing options are considered. The organization is a provider of e-commerce applications and web applications for SME's and (semi-) government organizations. The organization was founded 16 years ago, and currently comprises 48 employees of which 35 are developers. The organization is relatively young with ages of employees varying from 20 to 40 years.

All five sourcing options identified above are standard practice in this organization. However, the organization does not have a formal policy regarding software component sourcing decision criteria. Therefore, we expected that developers and managers are forced to contemplate the sourcing decision regularly, resulting in the building up of experience. In a sense, they can be considered as an expert group. We expected this would give us a wider and well informed range of answers. Since the organization had no formal policy, documentation was unlikely to provide relevant information. This also explains our reliance on interviews.

To constrain the interviewees into the sourcing decisions they actually make, rather than to trigger unsubstantiated perceptions and opinions, we focused at the decisions that had been made in the recent past on three specific projects. Within this organization we strived for maximum variation to promote diversity of results.

Thus recent projects were selected representing all the different sourcing options. The projects were each taken from different departments within the organization, to further increase the potential diversity of answers. Similarly, per project different stakeholders were interviewed, to account for role-based bias. Stakeholders having a role in sales, project management and software development, were selected. To increase response quality even further, only staff members with at least three years of experience were interviewed.

Within this setting a detailed design of the two parts of the interview was developed. During the first part, an open interview was conducted where the participants were encouraged to recollect the arguments actually used within the specific projects. The respondents were not shown the results of the literature study to prevent any unintentional bias. Respondents were asked to identify the components of the project. For each component identified they were asked:

- Were in your opinion other alternatives available?
- Did any colleague suggest other alternatives?
- On the basis of what criteria did you choose this option?

The interviews were recorded and crucial parts were transcribed. Using NVivo (Bazeley and Jackson, 2013) the results were organized and labelled. Subsequently the results were compared by the researchers with the list of criteria from the literature search, and matches and mismatches have been identified, of which the latter resulted in the identification of new additional criteria

The second part of the interview was more structured. The basis of this part of the interview is the list of criteria found in the literature. The goal is to see if these criteria identified in literature have been used or could have been used in practice. Also in this interview the relation with actual decision making practice will be maintained, so questions are again aimed at actual experience. Per criterion the following questions were asked:

- Have you used this criterion in an earlier decision?
- If yes:
  - Can you indicate what project this was?
  - To what extent has this criterion really contributed to the decision?
- If no (we did ask for opinions here):
  - Is this a plausible criterion?
  - Can you think of a project where this criterion would have relevant?

For the second part of the interviews transcription was not deemed necessary. Based on the recording, the discussions were sufficiently structured and clear.

The choice was made to do both parts of the interview in a single session. This had as an advantage that people remembered better what they said before and were therefore better able to connect what was mentioned in the first part, to the second part. This strengthened the results. It was also done for pragmatic reasons. It was easier to get participation this way. A drawback of course was that newly identified criteria could not now be tested across the participants.

Internal validity is fostered by a careful research design. Respondents were carefully selected and treated with respect. They were informed on the purpose of the project and were told their input was voluntary, would be treated anonymously and that they could, at any time, refuse an answer or stop their participation. They were also given the option to check our recordings and interpretations derived from their interview. Respondents were informed in advance about the purpose of the research and were also provided with definitions of the sourcing options. This allowed them to prepare the interview and also can prevent misunderstanding as to the object of discussion. This will increase the quality of the information obtained, and thus the validity of the research.

External validity is obtained by the 'factual' context maintained throughout the interviews. Results will show that in the particular organization some criteria have actually been used in the sourcing decision. Naturally, this does not imply relevancy for each and all other software organizations. But it does show that experienced practitioners have found them useful, hinting that others may value the use of explicit component sourcing criteria as well.

Reliability is again supported by the careful design of the interviews. This resulted in the development of an extended interview guide that allowed to a large degree repeatable interviews.

4 EXECUTION AND RESULTS

The literature study resulted in a list of 26 criteria (see table 1).

We selected three recent (within the last year) projects intending to cover all types of sourcing identified above. They were:

- P1: an e-commerce solution based on an internally developed e-commerce platform that uses open source software components and recycled in-house developed software.
- P2: an e-commerce solution based on the open source platform that uses open source software components, adapted open source software components, and in-house developed software.
- P3: an internal application framework that uses open source software components, closed source software components and in-house developed software.

Respondents were asked to identify the components in each project. Examples of components mentioned were Magento and Wordpress. This proved to be more complex than originally expected, resulting in some differences in components identified between the respondents. For project P1 the respondents identified three, five, and five components, resulting in the discussion of twelve components. For P2 the numbers were five, seven, and "two + others", also resulting in the discussion of twelve components. For P3 finally, the numbers were eight and nine resulting in discussion of fourteen components.

Table 1: Results from literature.

ID	Criterion
L01	Because the source code is publicly available the risk of stopping vendor support is reduced because there a possible to switch to another supplier (Ruffin and Ebert, 2004)
L02	Developing an application on a de facto standard API protects the application against changing supplier conditions (Ruffin and Ebert, 2004)
L03	The risk of having to provide compensation to the licensor for the breach of license, patent or proprietary rights (Ruffin and Ebert, 2004)
L04	The number of interactions between different components (Jha et al., 2014)
L05	The scale and complexity of software component (Daneshgar et al., 2013)
L06	Appropriate requirements - the extent to which the component standard meets user needs (Daneshgar et al., 2013)
L07	The number of discovered vulnerabilities (Schryen and Kadura, 2009)
L08	Lead time required to fix discovered vulnerabilities (Ruffin and Ebert, 2004)
L09	Reliability - maturity, fault tolerance and recoverability (Lawrence, 1996)
L10	Maintainability - analyzability, changeability, stability and testability (Lawrence, 1996)
L11	Effect of the software component on the availability of the system as a whole (Daneshgar et al., 2013)
L12	Flexibility in the use of the component (Daneshgar et al., 2013)
L13	Delivery time (Lim, 1994)
L14	Development costs (Lim, 1994)
L15	Life cycle / maintenance costs (Boehm and Bhuta, 2008; Favaro et al., 1998)
L16	The number of functional additions per release (Paulson et al., 2004)
L17	Freedom to adapt code (Chen et al., 2007)
L18	License of the component (Chen et al., 2007)
L19	Intellectual property (Daneshgar et al., 2013)
L20	Government requiring usage of specific accounting software (Daneshgar et al., 2013)
L21	Wish to maintain a broad technical vision across the entire product (Frakes, 2005)
L22	Wish to use knowledge and business expertise efficiently across projects (Frakes, 2005)
L23	Desire to systematically manage parts which allow flexible reaction to changing market conditions (Frakes, 2005)
L24	Availability of capable staff for development (Lim, 1994)
L25	Maintaining and keeping available reusable software components (Lim, 1994)
L26	Available financial means to organize re-use (Frakes, 2005)

An option here could have been to provide the component structure as an input for the interviews. However, in that case respondents could have been confronted with components they are not really familiar with. In many cases this would have resulted in additional answers. This would have decreased the reliability of the answers given. The results confirm that this indeed occurred during the interviews, with actually surprisingly little overlap between the components identified. This we feel, justified our design decision.

For each project, respondents were selected according to the roles specified above. Project P3 was an in-house project, so no related sales representative was available. Projects P1 and P2 were managed by the same project manager. This person was interviewed twice for the first part, once for each of the projects. The second part naturally only needed to be carried out once. In total, this resulted in eight interview results for part one and seven for the second part.

At this stage, we had the choice between compromising on the number of respondents or on the diversity of sourcing in the projects. We opted for an optimal diversity of sourcing options, feeling that sufficient interviews were left to give valid and reliable results.

The respondents had on average 8.2 years of experience of which 6.7 in their current organization, providing a solid basis of experience.

Table 2: Addition criteria found.

ID	Criterion	#
P01	experience with the software component within the organization	5
P02	availability of documentation	1
P03	interoperability and compatibility with plug-ins and / or frameworks	5
P04	the wish of the customer	6
P05	expected life of the software component	2
P06	software component is widely accepted by the community	4
P07	evaluation of the software component by the community	1
P08	Connect with market demand / increase commercial opportunities	3

For each interview we reserved four hours in a meeting room, so as to have sufficient time and to avoid being disturbed. On average, part one of the interview took slightly over half an hour, while the second part on average lasted for an hour. With some time required for the introduction and small rests between parts 1 and 2 and sometimes halfway part 2, the average duration was less than two hours. The

respondents had sufficient time to answer questions fully, contributing to the reliability of the answers.

The additional criteria found in the first part of the interviews can be found in table 2. In the column '#' is indicated the number of respondents who identified this criterion without prompting.

Of the 26 criteria identified in literature, eleven were also confirmed in this first part of the interviews. In table 3 the column 'part-1'shows the number of respondents that mentioned criteria (and an associated example from the project) that could be mapped to this list.

Table 3: Results interviews.

ID	Part-1	Part-2	
		based on usage	based on opinion
L01		3	5
L02			7
L03		3	3
L04	1	4	3
L05	3	5	1
L06	3	7	
L07		1	
L08			3
L09	3	6	1
L10	2	6	1
L11	3	5	2
L12	4	7	
L13		2	4
L14	6	6	1
L15	2	5	1
L16		3	2
L17		2	4
L18	5	6	1
L19		1	2
L20		2	5
L21		7	
L22		7	
L23		3	4
L24	2	4	3
L25		6	1
L26		5	

The second part of the interviews only looked at the criteria derived from literature, so no additional confirmation could be obtained for the criteria P1-P8. The results of the second part of the interviews can be found in the column 'part-2' table 3. The column 'based on usage' shows the number of respondents that recognized a criteria as one they had actually used in the past. An additional thirteen criteria from literature were confirmed here. In all cases an actual example was given by the respondents, demonstrating factual knowledge rather than speculation.

We also asked for opinions of respondents in case no actual usage took place. If they had not actually used the criterion they were asked if they found it plausible. The number of respondents who agreed with this can be found in the column 'based on opinion' of table 3.

When discussing criterion L7 (the number of discovered vulnerabilities) no fewer than five respondents stated that instead of the number of vulnerabilities the criterion should in fact consider their (potential) impact. One respondent out of these also provided an example of usage of this criterion in a recent project. This resulted in an unexpected ninth additional criterion:

P09: Impact of discovered vulnerabilities.

5 DISCUSSION AND CONCLUSIONS

In this paper we described research aimed at identifying criteria to support software component sourcing decisions. The literature study resulted in 26 potential criteria. Some criteria were mentioned by several authors, but in principle we saw limited overlap between the authors. This did not inspire confidence as to the completeness of this list. A more complete list would have shown more overlap.

This triggered design of an independent investigation, based on a series of in-depth interviews. Here experts from practice were asked, based on a recently completed project, to indicate criteria used in their sourcing decisions. This resulted in the identification of nineteen criteria, of which eleven could be matched to the list derived from literature and eight were new additions. A ninth addition emerged later from de interviews.

When discussing the quality of the resulting list of criteria we can first look at completeness. Naturally, the current list may be quite incomplete and further research is needed to establish a more complete list of commonly useful criteria. Nine new additions to a list based on the experience of just a single company does suggest that saturation has as yet not been achieved. We are likely to find more when more companies are included in the research.

On the other hand, by combining literature and practice in this way it would seem that at least the most obvious, and maybe then also the most important criteria, will have been identified. It must

be noted that the research conducted only looked at relevance, not at degree of importance.

Apart from completeness we predominantly looked at relevance of the criteria that were identified. There a more positive picture emerges. The nine new criteria have been found without prompting and have been used in practice for a concrete sourcing decision within the target organization. That implies that they are relevant and other organizations can consider using them.

Likewise, eleven criteria emerged from the interviews, which could be easily mapped on the literature. For these a similar degree of confidence can be expressed. Again these were found without prompting and have already been used in a sourcing decision practice. And they also have backing from literature.

In the second part of the interviews we used the list resulting from literature as input to the interviews. Out of the fifteen remaining criteria, for thirteen criteria examples were provided that they had been used in an actual decision process. The evidence can be considered slightly less strong since the respondents required prompting for these criteria but still examples of usage could be given. It is reasonable to conclude that these criteria are also relevant.

That leaves two criteria for which no actual usage could be identified. However, L02 (Developing an application on a de facto standard API protects the application against changing supplier conditions) was seen by all seven respondents to be a plausible criterion nonetheless. This remarkable consensus gives no evident reason to dismiss this criterion. L08 (lead time required to fix discovered vulnerabilities) is also confirmed three times. All in all, there are reasons to qualify the entire result as at least 'plausible'.

Furthermore, the initial list presented in this study is rather unrefined and needs additional processing. Many criteria are overlapping and differ in the level of abstraction and aggregation. E.g. criterion P04 rather broadly states the importance of "customer wishes". This is a more abstract formulation of the very specifically formulated L20 (Government requiring usage of specific accounting software). L07, L08 and P09 all somehow focus on vulnerabilities. L03 and L18 both consider license issues. Because of this, the current set of criteria cannot be seen as a set of independent criteria. Some further classification is required. We decided against doing so for the results of the literature study for two reasons. One because the number of criteria resulting was manageable and the other because we did not want to run a risk of changing information by our interpretations. The current list can be classified further, but we decided to wait till additional criteria have been identified.

Obviously, further research will be needed to further validate this set of criteria and to add more results and insights from practice. An ongoing effort is required to discover more potentially useful criteria, which may hopefully result in some sort of saturation. After that the resulting list can be classified in a more coherent and manageable form.

There is also the interesting aspect of (relative) degree of importance of criteria. This is probably very much context dependent and therefore local assessment will be needed to make a "common criteria list" operational in decision making practices in software component sourcing. This would open up a new line of research in which the decision making process of the way in which software components are sourced comes into focus.

REFERENCES

Bazeley, P., and Jackson, K. (Eds.). 2013. *Qualitative data analysis with NVivo.* Sage Publications Limited.

Boehm, B., and Bhuta, J. 2008. Balancing opportunities and risks in component-based software development. *IEEE Software*, 25 (6), 56-63.

Chen, W., Li, J., Ma, J., Conradi, R., Ji, J., and Liu, C. 2007. A survey of software development with open source components in China's software industry. *Lecture Notes in Computer Science,* Vol. LNCS 4470, pp. 208-220.

Cortellessa, V., Marinelli, F., and Potena, P. (2008). An optimization framework for "build-or-buy" Decisions in software architecture. *Computers and Operations Research*, 35 (10), 3090 - 3106.

D'Souza D. F. and Wills A.C., 1997. *Objects, Components, And Frameworks with UML – the Catalysis Approach*, Addison-Wesley, Reading, Mass.

Daneshgar, F., Low, GC, and Worasinchai, L. 2013. An investigation of "build vs. buy "decision for software acquisition by small to medium enterprises. *Information and Software Technology*, 55 (10), 1741-1750.

Favaro, JM, Favaro, KR, and Favaro, PF. 1998. Value based software reuse investment. *Annals of Software Engineering,* 5 (1), 5-52.

Frakes, WB 2005. Software reuse research: status and future. *IEEE Transactions on Software Engineering*, 31 (7), 529-536.

Jadhav AS, and Sonar RM 2011. Framework for evaluation and selection of the software packages: A hybrid knowledge based system approach. *Journal of Systems and Software*,. 84 (8) 1394-1407.

Jha, PC, Bali, V., Narula, S., and Kalra, M. 2014. Optimal component selection based on cohesion and coupling

for component-based software system under build-or-buy scheme. *Journal of Computational Science*, 5 (2), 233-242.

Kakarontzas, G. Constantinou, E., Ampatzoglou, A., and Stamelos, I. 2013. Layer assessment of object-oriented software: A metric facilitating white-box reuse. *Journal of Systems and Software*, 86 (2), 349-366.

Lawrence, S. 1996. Software Quality: The Elusive Target. *IEEE Software*, 13 (01), 12-21.

Li, J., Conradi, R., Slyngstad, OPN, Bunse, C., Torchiano, M., and Morisio, M. 2006. An empirical study on decision making in off-the-shelf component-based development. *Proceeding of the 28th International Conference on Software Engineering - ICSE '06*, 897-900.

Lim, W. 1994. Effects of reuse on quality, productivity, and economics. *IEEE Software*, 11 (5), 23-30.

Morisio, M. Seaman, C., Basili, V., Parra, A., Kraft, S., and Condon, S. 2002. COTS-based software development: Processes and open issues. *Journal of Systems and Software*, 61, 189-199.

Paulson, JW, Succi, G., and Eberlein, A. 2004. An empirical study of open-source and closed-source software products. *IEEE Transactions on Software Engineering,* 30 (4), 246-256.

Ruffin, C., and Ebert, C. 2004. Using open source software in product development: A Primer. *IEEE Software*, 21 (1), 82-86.

Schryen, G., and Kadura, R. 2009. Open source vs. closed source software: towards measuring security *Proceedings of the 2009 ACM Symposium on*, 2016 - 2023.

Knowledge Management and e-Learning Integration Model (KMELI)

Janis Judrups
Baltijas Datoru Akademija, Tallinas 4, Riga, Latvia
janis.judrups@bda.lv

Keywords: Knowledge Management, e-Learning, Integration Model.

Abstract: The article offers a model for knowledge management and e-learning integration (KMELI). The purpose of this model is to support the development of human resources in business environment and use learning as a common field for both these disciplines, with a particular emphasis on the determination of learning needs on the level of the organisation and the employee. The instructional design approach-based methodological framework that describes in detail the activities conducted in each phase is offered for the practical implementation of the model. It is important that, before training development is started, an initial analysis takes place, in order to separate learning needs from those that cannot be met with the help of training.

1 INTRODUCTION

Knowledge management (KM) and e-learning (EL) are developed as recognized, self-contained disciplines for years. By shifting focus on knowledge as the main resource of organization, these disciplines are gaining more and more interest. With further development, synergistic relationships should increase between knowledge management an e-learning (Liebowitz and Frank, 2011). Some of these relationships are quite evident, because both disciplines:

- Deal with knowledge capture, sharing, application and generation;
- Have important technological components to enhance learning;
- Contribute to building a continuous learning culture;
- Can be decomposed into learning objects.

Several conceptual, technological, organizational and content barriers are hindering close integration of knowledge management and e-learning (Brown et al., 1989, Brusilovsky and Vassileva, 2003, Benmahamed et al., 2005, Dunn and Iliff, 2005, Maier and Schmidt, 2007). For example, workplace of a knowledge worker is fragmented: separated work, knowledge and learning space; KM and EL use separate ICT systems and different technologies (Ley et al., 2005); amount of guidance that KM and EL provide for learner is not appropriate; KM and EL have limited and isolated consideration of context (Schmidt, 2005); KM materials are missing interactivity (Yacci, 2005).

By overcoming integration barriers we may expect clear benefits for both disciplines and increased quality, convenience, diversity and effectiveness within an organization (Yordanova, 2007, Sammour and Schreurs, 2008, Islam and Kunifuji, 2011).

There are several theoretical knowledge management an e-learning integration models described in literature (Woelk and Agarwal, 2002, Schmidt, 2005, Sivakumar, 2006, Maier and Schmidt, 2007, Mason, 2008, Islam and Kunifuji, 2011, Ungaretti and Tillberg-Webb, 2011). Analysis of these models shows several integration ways and approaches, however, these models are not implemented in production environment and lack necessary technical specification and application support (Judrups, 2015a). As result of specific organizational goals and needs models employ different adaption and integration approaches (Judrups, 2015b). The more general approach is to base integration on common ground, which was identified as learning.

The goal of the study was to develop a solution that would allow a training centre to be efficient in ensuring the development of employees to accomplish the objectives of the organisation and complete work tasks in business environment.

Unfortunately, none of the models described in

Judrups, J.
Knowledge Management and e-Learning Integration Model (KMELI).
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 588-593
ISBN: 978-989-758-187-8

the literature was of practical use in a situation like this. This is why a new knowledge management and e-learning integration model (KMELI) was created. For the practical implementation of the model, the methodology (implementation framework) based on instructional design approaches was designed. Thus, the goal of this article is to describe the KMELI model developed, as well as its implementation framework.

2 CONTEXT OF THE MODEL

The context of the development of the model was based on a broader study of human resource and business management processes and their interaction, which created a competence-based human resource management framework (Judrups, 2015a). This is why the KMELI model must comply with the following approaches:

- employee development uses a competence-based approach;
- competence assessment uses e-learning-based solutions;
- personalised development plans are composed for employee development;
- development solutions used are described through competences and summarised in a development solution catalogue.

The following requirements were set for the KMELI model:

- meet the learning needs of the organisation;
- meet the formal and informal learning needs of the employees with the use of KM and EL;
- support automated competence assessment;
- support employee competence profile and competence gap use;
- support the use of personalised employee development plans;
- support the use of development solutions described through competences: resource creation, publication, implementation.

It is intended that these requirements and approaches will be elaborated more on further stages of the study; therefore, the KMELI model must be developed as sufficiently conceptual and general.

3 BACKGROUND OF THE MODEL

Training is the basis of both the knowledge management and e-learning, because both these disciplines are crucial components of training processes. The interaction and the specific approaches of KM and EL help achieve the learning goals set by the organisation (Ungaretti and Tillberg-Webb, 2011).

The understanding of KM and EL processes can be considered and compared as value chains of both these disciplines (Wild et al., 2002). The value chains in both the disciplines comprise four sequential processes that can be divided into two stages: (1) identification of needs and goals; (2) design, development, implementation (see Figure 1).

A comparison of the value chains of knowledge management and e-learning shows close relations between these disciplines. The commitment of the organisation towards e-learning is directly related to the first two processes in the knowledge management value chain: that is, the necessity to identify the strategic knowledge needs of the organisation and the lack of required knowledge. The last two processes in the KM value chain (the elimination of knowledge gap, and the distribution and use of the knowledge obtained) coresponds with the last three processes in the EL value chain. Proper development of the content and the learning approach, followed by the implementation of e-learning, allows to eliminate knowledge gap and distribute knowledge in the organisation, boosting its development and improving its competitiveness (Wild et al., 2002).

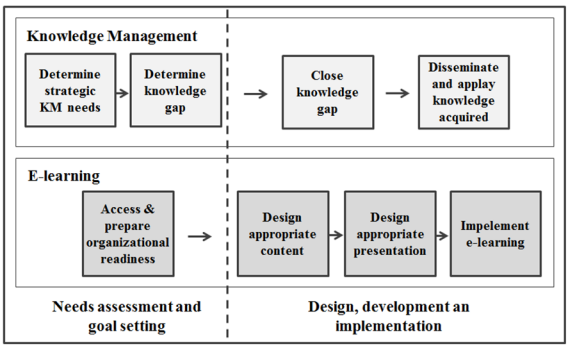


Figure 1: KM and EL value chain comparison.

KM and EL both serve the same purpose: improving learning and competence development in the organisation. However, they use two different perspectives. KM uses the organisation-level perspective, in order to avoid insufficient sharing of information among the employees of the organisation. On the other hand, e-learning emphasises the perspective of the individual, focusing on obtaining individual knowledge (Ras et al., 2005).

Proper selection of metrics and their consistent use allows confirming the accomplishment of the

goals set. The main problem lies not in finding the quality standards itself, but in choosing the most appropriate ones from the broad selection of standards available (Ehlers, 2005).

4 DESCRIPTION OF THE MODEL

The KMELI model demonstrates the integration of knowledge management and e-learning with learning as the common aspect of both these disciplines (see Figure 2).

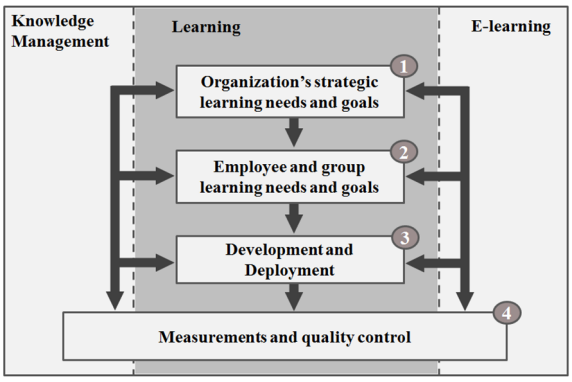


Figure 2: KMELI KM and EL integration model.

The organisation learning cycle begins with the identification of knowledge needs and goals on the strategic level of the organisation (1). This allows to strengthen the traditionally individual aspect of e-learning and to provide a broader learning context by connecting learning results with the strategic goals and objectives of the organisation.

The learning needs and objectives are further specified on the level of individual employees and groups of employees (2). The acknowledgement of the context of the employee (personal learning traits, professional functions, tasks and processes, etc.) allows to personalise the learning solution and to involve the employee better in the learning process, helping the employee be more successful in achieving the results of the learning.

During the development, implementation and execution of the learning (3), the learning is prepared and conducted, ensuring the acquisition, distribution and of the relevant knowledge in the organisation. All the three stages mentioned above are further subjected to quality control with the help of the metrics selected (4). In the model, this process is deliberately shown as a block that comes out of the common part of the integration between KM and EL (learning), because the process of quality control must ensure successful work of all the KM and EL implemented. It is important that the process of quality control allows both ensuring control and introducing correction on all the three levels. This is one of the aspects that will define the quality standards to be used in a practical implementation of the model.

5 ANALYSIS OF THE MODEL

The analysis of the KMELI model confirms that it complies with all the requirements set for its development:

- The learning needs of the organisation are identified on the first step of the model (1) (see Figure 2). The learning and knowledge needs are related to the strategic goals of the organisation, providing them with the context of the organisation and allowing its employees to understand better the goals of learning.
- The formal and informal training of employees is planned for the second step of the model (2). It is coordinated with the strategic goals of the organisation. This step provides for the use of individual development and training plans, particularly for longer-term training and for developing competencies that are more difficult to learn. The acquisition of minor knowledge necessary for daily work may not appear in individual development plans, because it can take place with the help of knowledge management techniques, such as informal training, tips from experienced colleagues, use of an archive for the training completed etc.
- The automated competence assessment can be accomplished with the use of e-learning knowledge assessment tools, which are based on various tests and agent software that monitors the employee during work hours. The results obtained would then be submitted and processed for the employee's competence profile. The evaluation of quality and training results (4) allows confirming the accomplishment of the goals of the training, and the acquisition of the competences planned. This information would then be registered in the competence profile of the user, decreasing the competence gap and updating as needed the further development plan.
- All the knowledge and training objects used in training can be described with the help of

competences as development resources and registered in the development solution catalogue. In order to use these resources successfully, it is necessary to create or repurpose a small, self-contained module in a way that creates a mutual content-based and pedagogic connection among them. Competences are used to describe the training goal of these modules and the prerequisite knowledge for the training (Schmidt, 2005).

It can be observed that, at its core, the KMELI model has an organisation of learning processes with a distinct emphasis on connecting the learning objectives with the general strategic goals of the organisation (1), on taking into account the specific needs and contexts of the employees (2), on quality control applied throughout the process, and on achieving the goals set (see Figure 2).

Although the model is based on the knowledge management and e-learning disciplines, this aspect is not reflected significantly in the organisation of the learning processes. Therefore, the use of the model can be expanded to the entire learning process and applied according to the needs of the organisation.

The learning needs of work groups and employees may not arise directly from the cascading of the strategic goals of the organisation and its needs. These needs can be related to the performance and performance ratings of specific employees. These needs would, in fact, begin being met on Stage 2, while the strategic goals would allow to confirm that the work done is necessary and to provide a broader context for the training.

6 RESULTS AND DISCUSSIONS

Taking into account the analysis and the conclusions, it is possible to determine the main principles of the KMELI model:

- KMELI demonstrates the integration of nowledge management and e-learning with training as the common aspect of both these disciplines;
- The identification of learning needs and goals begins at the level of the organisation;
- The learning needs and goals of the employee at the individual level and the level of work groups are specified and put into the contexts of the employee;
- The development, implementation and execution of training provides the acquisition, distribution and use of knowledge in the company;
- The metrics and quality control on all the three stages ensure the improvement of processes and products, as well as the attainment of results.

Practical implementation of the model developed requires methodology, thus a KMELI implementation framework was developed. It clarifies the activities conducted on each KMELI phase and serves as a detailed example for learning processes at the organisation. The framework helps in the introduction and development of such processes at the organisation. The main target audience of the KMELI framework are organisations that provide their employees, clients and partners with training. The organisations that provide training to external clients may need to modify the training objective identification processes.

A KMELI framework must be able to answer the following questions:

- How are the strategic learning goals and needs of the organisations defined?
- How are the learning goals and needs of employees and their groups defined?
- How is training developed and implemented?
- What are the quality control mechanisms and what metrics are to be used?

The KMELI framework tries to answer the questions that are usually resolved with help of the instructional design. The instructional design is a systematic process that is used to turn teaching and training principles into traininig materials and activities (Smith and Ragan, 1993). The development of training is based on five stages: analysis, design, development, implementation, evaluation. This general approach is called the ADDIE model (see Figure 3), customised variants of which are usually created for practical use in organisations (Molenda, 2003).

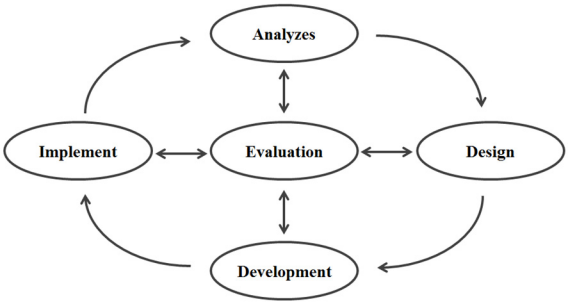


Figure 3: ADDIE dynamic model (adopted from (Schufletowski, 2002)).

During the first stage of the KMELI model, the learning needs of the organisation are determined. This is similar to the ADDIE model, in which the analysis stage is used to study the needs and the environment. Such analysis often employs need evaluation or performance evaluation techniques. In both cases, a list of the needs of the organisation can be obtained, although only a part of these needs would be directly related to the needs of learning (Molenda and Russell, 2006).

A part of the solutions to performance problems would not be related to the use of training at all, and in most cases training will only be a part of a bigger solution. During the initial analysis, the learning needs are separated from other performance problems. The development of training is conducted to satisfy only the learning needs. Therefore, it is practical to introduce the initial analysis stage of the KMELI implementation framework, which will determine the learning needs of the organisation and then transfer it further for the instructional design process (see Figure 4).

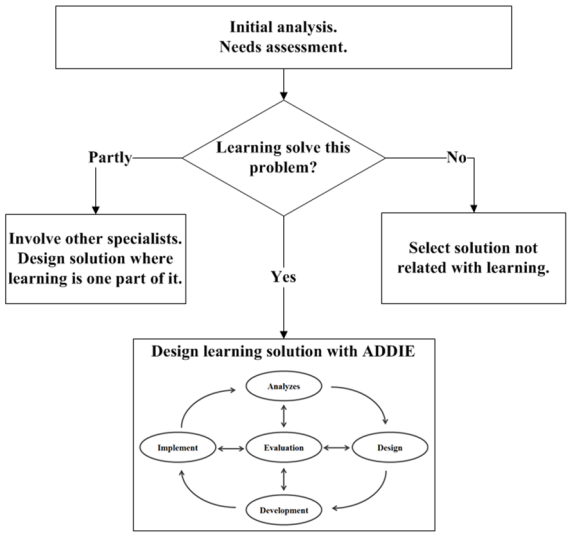


Figure 4: Initial analysis and learning needs assessment.

As a result, the KMELI implementation framework can be divided into six stages: the initial analysis and the five stages of the ADDIE model (analysis, design, development, implementation, evaluation). It is clear that, in practice, the initial analysis will be closely related to the following analysis stage, although the decision on the necessity and justifiability of training will be a crucial milestone.

7 CONCLUSIONS

The KMELI model provides a theoretical foundation for creating a practically usable knowledge management and e-learning integration solution. For the practical use of the model, the methodology – implementation framework based on instructional design approaches was designed. It provides a detailed description of the activities conducted on each of the stages of the model. It is important that, before training development is started, an initial analysis take place, in order to separate learning needs from those that cannot be met with the help of training.

Further study requires that the model and its framework are verified in practice. Successful verification results will allow their further use in the development of a functioning knowledge management and e-learning integration solution.

ACKNOWLEDGEMENTS

Research is part of project „Competence Centre of Information and Communication Technologies" run by IT Competence Centre, contract No. L-KC-11-003, co-financed by European Regional Development Fund.

REFERENCES

Benmahamed, D., Ermine, J.-L., Tchounikine, P., 2005. From MASK Knowledge Management Methodology to Learning Activities Described with IMS – LD. In K.-D. Althoff, A. Dengel, R. Bergmann, M. Nick, and T. Roth-Berghofer, eds. *Third Biennial Conference, WM 2005, Kaiserslautern, Germany, April 10-13, 2005, Revised Selected Papers*. Springer Berlin Heidelberg, 165–175.

Brown, J. S., Collins, A., Duguid, P., 1989. Situated Cognition and the Culture of Learning. *Educational Researcher*, 18 (1), 32–42.

Brusilovsky, P., Vassileva, J., 2003. Course sequencing techniques for large-scale web-based education. *International Journal of Continuing Engineering Education and Lifelong Learning*, 13 (1-2), 75–94.

Dunn, P., Iliff, M., 2005. Learning Light At Cross Purposes Why e-learning and knowledge management don't get along.

Ehlers, U. D., 2005. *Quality in e-learning: use and dissemination of quality approaches in European e-learning: a study by the European Quality Observatory*. Luxembourg: Office for Official Publications of the European Communities.

Islam, M., Kunifuji, S., 2011. Adopting Knowledge Management in an E-Learning System: Insights and Views of KM and EL Research Scholars. *Knowledge Management & E-Learning*, 3 (3), 375–398.

Judrups, J., 2015a. Analysis of Knowledge Management and E-Learning Integration Models. *Procedia Computer Science*, 43, 154–162.

Judrups, J., 2015b. Analysis of Knowledge Management and E-Learning Integration Approaches. In S. Hammoudi, L. A. Maciaszek, and E. Teniente, eds. *{ICEIS} 2015 - Proceedings of the 17th International Conference on Enterprise Information Systems, Volume 2, Barcelona, Spain, 27-30 April, 2015*. SciTePress, 451–456.

Ley, T., Lindstaedt, S., Albert, D., 2005. Supporting competency development in informal workplace learning. In *Lecture Notes in Artificial Intelligence - Professional Knowledge Management: Third Biennial Conference, WM 2005, Revised Selected Papers*. Kaiserslautern, Germany: Springer Berlin Heidelberg, 189–202.

Liebowitz, J., Frank, M. S., 2011. The Synergy between Knowledge Management and E-Learning. In J. Liebowitz and M. S. Frank, eds. *Knowledge management and E-learning. Innovations in education and teaching international*. CRC Press, 3–10.

Maier, R., Schmidt, A., 2007. Characterizing knowledge maturing: A conceptual process model for integrating e-learning and knowledge management. In *4th Conference on Professional Knowledge Management. Experiences and Visions*. Berlin: GITO-Verlag, 325 – 333.

Mason, J., 2008. A Model for Exploring a Broad Ecology of Learning and Knowing. In *Supplementary Proceedings of the 16th International Conference on Computers in Education, Asia-Pacific Society for Computers in Education (APSCE)*. Taipei, 194–203.

Molenda, M., 2003. In search of the elusive ADDIE model. *Performance Improvement*, 42 (5), 34–36.

Molenda, M., Russell, J. D., 2006. Instruction as an Intervention. In J. A. Pershing, ed. *Handbook of Human Performance Technology Improvement*. Pfeiffer, 335 – 369.

Ras, E., Memmel, M., Weibelzahl, S., 2005. Integration of e-learning and knowledge management – barriers, solutions and future issues. In *Professional Knowledge Management. Third Biennial Conference, WM 2005, Kaiserslautern, Germany, April 10-13, 2005, Revised Selected Papers*. Berlin: Springer Berlin Heidelberg.

Sammour, G., Schreurs, J., 2008. The role of knowledge management and e-learning in professional development. *Knowledge and Learning*, 4 (5), 465–477.

Schmidt, A., 2005. Bridging the gap between knowledge management and e-learning with context-aware corporate learning. In *Professional knowledge management. Third Biennial Conference, WM 2005, Kaiserslautern, Germany, April 10-13, 2005, Revised Selected Papers*. Springer Berlin Heidelberg, 203–213.

Schufletowski, F. W., 2002. *AIR Force Handbook 36-2235 Volume 1*.

Sivakumar, S. C., 2006. E-Learning for Knowledge Dissemination. In D. Schwartz, ed. *Encyclopedia of knowledge management*. Idea Group, 152–160.

Smith, P. L., Ragan, T. J., 1993. *Instructional Design*. New Yourk: Merrill.

Ungaretti, A. S., Tillberg-Webb, H. K., 2011. Assurance of Learning: Demonstrating the Organizational Impact of Knowledge Management and E-Learning. In J. Liebowitz and M. S. Frank, eds. *Knowledge management and E-learning. Innovations in education and teaching international*. CRC Press, 41–60.

Wild, R. H., Griggs, K. A., Downing, T., 2002. A framework for e-learning as a tool for knowledge management. *Industrial Management & Data Systems*, 102 (7), 371–380.

Woelk, D., Agarwal, S., 2002. Integration of e-Learning and Knowledge Management. In *World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education*. 1035–1042.

Yacci, M., 2005. The Promise of Automated Interactivity. In K.-D. Althoff, A. Dengel, R. Bergmann, M. Nick, and T. Roth-Berghofer, eds. *Professional Knowledge Management SE - 24*. Springer Berlin Heidelberg, 214–221.

Yordanova, K., 2007. Integration of Knowledge management and E-learning – common features. *CompSysTech 07 Proceedings of the 2007 international conference on Computer systems and technologies*.

Fuzzy Clustering based Approach for Ontology Alignment

Rihab Idoudi[1,2], Karim Saheb Ettabaa[2], Kamel Hamrouni[1] and Basel Solaiman[2]
[1]*Université Tunis ElManar, Ecole Nationale d'Ingénieurs de Tunis, Tunis, 1200, Tunisia*
[2]*Laboratoire ITI, Telecom Bretagne, Brest, 29238, France*

Keywords: Fuzzy C-Medoid, Ontology Aligning, Semantic Similarity, Similarity Measures.

Abstract: Recently, several ontologies have been proposed for real life domains, where these propositions are large and voluminous due to the complexity of the domain. Consequently, Ontology Aligning has been attracting a great deal of interest in order to establish interoperability between heterogeneous applications. Although, this research has been addressed, most of existing approaches do not well capture suitable correspondences when the size and structure vary vastly across ontologies. Addressing this issue, we propose in this paper a fuzzy clustering based alignment approach which consists on improving the ontological structure organization. The basic idea is to perform the fuzzy clustering technique over the ontology's concepts in order to create clusters of similar concepts with estimation of medoids and membership degrees. The uncertainty is due to the fact that a concept has multiple attributes so to be assigned to different classes simultaneously. Then, the ontologies are aligned based on the generated fuzzy clusters with the use of different similarity techniques to discover correspondences between conceptual entities.

1 INTRODUCTION

As the study on data engineering actively progresses, knowledge management constitutes, nowadays, a primordial problematic, where the challenge relies on resolving the knowledge capitalization problem by improving knowledge merge and share. In this context, ontologies are introduced as a potential mean for conventional knowledge modeling for any given complex domain (Idoudi et al., 2014). In practice, several ontologies within the same domain are developed independently by different communities. Consequently, to date, the popularity of ontologies is rapidly rising, and the amount of available ontologies remains increasing. Thus, in case of knowledge sharing, it is crucial to establish interoperability between those ontologies to handle the semantic heterogeneity problem (Hamdi and Safar, 2009). Several ontology engineering processes are assuming this task, mainly the ontology alignment. This area of research has resulted in numerous studies (Fernández et al., 2012); (Shvaiko and Euzenat, 2005); (Qiu and Liu, 2014). Nevertheless, most of those approaches fail spectacularly to capture adequate correspondences when dealing with large ontologies of extremely different levels of granularities (Duan et al., 2011). This is due to the size and monolithic nature of these large ontologies. In this paper, we direct our attention to explore ways of ontology aligning. We therefore propose and evaluate a new, more efficient, fuzzy clustering-based approach. The main objective of adopting the fuzzy clustering is that it contributes to optimal organization of the ontological structure and it ensures that all the resulting clusters are concise enough to avoid any loss of information. The alignment process is based on three main steps; first the candidate ontology is clustered into concise clusters with estimation of medoids to the different generated clusters. Thus, we propose a semantic distance for clustering analyze. Second, clusters of both ontologies are aligned by means of their medoids using the semantic similarity to determine similar clusters. Once the pairs of similar clusters are retained, the third step consists on aligning the correspondent entities. Although, several clustering based alignment methods have been proposed, our approach is characterized the use of fuzzy clustering to avoid information loss when ontologies clustering. Moreover, our method uses the medoid notion to determine similar blocks, contrarily to exisitng method which consist on parsing the whole cluster's entities to conclude similar ones. The rest of the paper is organized as follows: in the next section, we introduce some related works. In Section 3, we propose our algorithm for ontology fuzzy clustering. In Section 4, we present an alignment method. In Section 5, we show some initial

Idoudi, R., Ettabaa, K., Hamrouni, K. and Solaiman, B.
Fuzzy Clustering based Approach for Ontology Alignment.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 594-599
ISBN: 978-989-758-187-8

experimental results to demonstrate the efficiency of the method.

2 RELATED WORK

Thus, in order to perform ontology alignment process, several researchers have been interested to perform clustering techniques over ontologies. In (Algergawy et al., 2011), the author proposed a clustering approach based on structural nodes similarity. Therefore, each cluster of the source ontology has to be aligned with only one subset of the target ontology. In (Seddiquia and Aono, 2009), the approach starts by anchoring, a pair of "look-alike" neighbors concepts to be aligned. The method outputs a set of alignments between concepts within semantically similar subsets. The authors in (Hu et al., 2006) address the problem of aligning large class hierarchies by introducing a partition-based block approach. The process is based on predefined anchors and uses structural and linguistic similarities to partition class hierarchies into small blocks. The COMA++ system presented in (Massmann et al., 2011) consists on partitioning large ontologies by using relatively simple heuristic rules. It starts by transforming ontologies into graphs. Then, clustering algorithm is applied to partition the graphs into disjoint clusters. To determine similar clusters, the aligning process uses limited information about the cluster, which results in less alignment quality. In (Hu et al., 2008), starting from small clusters, Falcon-AO system merges progressively clusters together. The alignment process, exploits the whole cluster information to determine clusters pairs having higher proximity. This proximity is based on anchors. The more these clusters share anchors, the more similar they are. A structural clustering method based on network analysis was proposed in (Schlicht and Stuckenschmidt, 2008). The latter produces, in a consuming time, an important number of too small modules (which may affect the concept's overall context). Authors in (Wang et al., 2011) use two types of reduction anchors to align ontologies. In order to predict ignorable similarity calculations, positive reduction anchors use the concept hierarchy while negative reduction anchors use locality of matching.

3 ONTOLOGY FUZZY CLUSTERING

In this section, we present our method for ontology fuzzy clustering using the FCMdd algorithm over ontology concepts. The use of fuzzy clustering is justified by the fact that a concept has multiple attributes so to be assigned to different classes simultaneously. Second, the use of fuzzy clustering may significantly reduce the loss of information while concept's clustering.

3.1 The FCMdd Algorithm

FCMdd clustering technique represents a variant of the FCM technique applied over relational data. Likewise, the FCMdd allows computing membership degrees of concepts to different clusters as well as medoids which represent the representative data of the clusters. These fuzzy clusters groups semantically close concepts, where the membership to each cluster is not deterministic but rather ranges in the unit interval [0, 1]. It is worth to note that we are interested only in this work to concepts $X = \{x_1, \dots, x_n\}$, while relationships $R(x_i, x_j)$ are used to determine similarity in the clustering task. FCMdd is an iterative algorithm which tends to minimize this objective function:

$$J_M(X, V) = \sum_{i=1}^{c} \sum_{k=1}^{n} (u_{ik})^m \, d(x_k - v_i)^2 \qquad (1)$$

Let $X = \{x_1, \dots, x_n\}$ be a set of ontology concepts where n is the number of nodes in ontology, d()denotes the semantic distance between two concepts of X. The set $V= \{v_1, \dots, v_c\}$ represents a subset of X with cardinality c(number of clusters); it represents the medoids set of the clusters, u_{ik} is the membership degree of element x_k to cluster i with $\sum_{i=1}^{c} u_{ik} = 1$. m is the fuzziness parameter of the resulting clusters where $m > 1$.

The membership degree is defined as well:

$$u_{ik} = \frac{\left(\frac{1}{d(x_k, v_i)}\right)^{\frac{1}{m-1}}}{\sum_{j=1}^{c}\left(\frac{1}{d(x_k, v_j)}\right)^{\frac{1}{m-1}}} \qquad (2)$$

Specifically, each cluster will be represented by a medoid. The latter represents the concept that has the minimal average distance with respect to the others. Formally the medoid of cluster C, where $v_i, c_j \varepsilon C$; w. r. t. the semantic distance d(.):

$$v_i = \arg\min_{c_i \epsilon C} \left(\frac{1}{|C|} \sum_{j=1}^{n} d(v_i, c_j)\right) \qquad (3)$$

The medoids designate the concepts minimizing the distance to the other members of the cluster e.g in the alignment step; those prototypes may intentionally speed-up the task of searching closest clusters. Finally,

a specific similarity measure for concepts is needed. The latter is presented in the next section.

3.2 New Semantic Distance

Intuitively, we assume that two concepts are particularly closer while the distance between them is minimal; to estimate the distance, we consider the relational context of a concept. The idea is to define for each concept a relational context that reveals the entities to which the concept is related in the ontology. The context must hold the knowledge to express the circumstances of a concept, its role in the ontology and its use cases. For this, we consider both kinds of relationship: First, the subsumption relation that gives information about concepts subsumed by the concept of interest or the concept that subsume it. Second, we consider the object property relation which reveals the connected concepts. Given C the set of concepts in ontology, R the set of relations including the subsumption and object property relations, the relational context of a concept $c \varepsilon C$ is given by:

$$Cont(c) = \{c_i | (c, c_i) \varepsilon R \cup \{c\}\}$$

Figure1 gives an example of the relational context of the concept 'Calcification' in the mammographic ontology, where we can see that it is related according to subsumption relation with {Micocalcification, Maco-calcification, Lesion} and according to object-property relation with {Cyst, Mass, Opacity}, then, we can define the relational context of the concept 'Calcification' as {Calcification, Mico-calcification, Macocalcification, Lesion, Cyst, Mass, Opacity}.

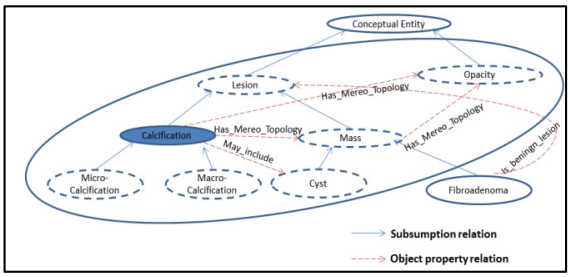


Figure 1: Relational context of the concept 'Calcification'.

Given two concepts c_i and c_j, the distance $d(c_i, c_j)$ based on relational context between them is given as well:

$$d(c_i, c_j) = 1 - \left(2.\frac{|C(c_i) \cap C(c_j)|}{|C(c_i)| + |C(c_j)|}\right) \quad (4)$$

$|C(c_i) \cap C(c_j)|$ Represents the number of common elements between the contexts of c_i and c_j.

3.3 The Fuzzy Clustering Algorithm

Algorithm 1 illustrates the FCMdd based ontology clustering based algorithm. The inputs of the algorithm are m: the fuzziness parameter, c: the number of clusters (determined by application requirement.) as well as the membership degrees and medoids set initialisation. The output of the algorithm is a set of clusters with correspondant medoids and the membership degrees of the concepts to the different clusters. It is worth to note, that the use of medoids is particularly important for a more flexible representation of clusters. Moreover, it helps to speed up the task of determining similar clusters between candidate ontologies.

Algorithm 1: FCMdd based Ontology Clustering.

Input: $X = \{x_1, \ldots, x_n\}$: set of Ontology's concepts,
c : Number of clusters ,
m: Fuzziness parameter ,
V= $\{v_1, v_2, \ldots, v_c\}$: set of medoids
MaxIter
Output: C_c set of clusters of concepts
Begin
Initialize the membership degrees u_{ik}
for $i = 1, \ldots, c$, $k = 1, \ldots, n$
Initialize the set of medoids V= $\{v_1, v_2, \ldots, v_c\}$
Repeat
Compute membership degrees u_{ik} for $i = 1 \ldots c$
and $k = 1 \ldots n$ According to (2);
Update v_i; i=1...c according to (3);
$V_{ancien} = V$
Iter=iter+1
Until ($V_{ancien} = V$ // convergence or iter=MaxIter)
Return C_c

4 CLUSTERS ALIGNMENT

In this section, we present our approach for clusters alignment. The input of the algorithm is the set of clusters correspondent to the source and target ontologies to be aligned. The idea is to compare both sets of clusters using the predefined medoids since these prototypes give a sketch of the clusters content. Thus matching medoids is helpful for users to understand the correspondences between clusters. The comparison is based on the use of semantic similarity. For each source cluster, we compute the semantic similarity of its medoid with the target medoids. The most similar medoids are retained to compare their respective clusters's entities in the next step. The semantic similarity computation uses an external resource to compute the similarity value. In this method, we have used the WordNet thesaurus

which groups words (nouns, verbs, adjectives) into sets of synonyms called synsets. The latte contains all the terms denoting a concept. They are linked by semantic relationship such as generalization or specialization relationship. The similarity between two synsets A and B of the two concepts c_1, c_2 is computed as well:

$$sim_{semantic}(c_1, c_2) = =\max(A \cap B/A \cup B) \quad (5)$$

Once we have determined the couples of clusters deemed to be similar. We move from supervising predefined matched class pairs to their correspondent entities. At this step, we assume that as long as two medoids of source and target clusters are semantically close, their respective clusters have to be aligned. It is then carried to fully align elements inside retained similar clusters with the use different similarity measures such as the syntactic similarity and the structural similarity. The syntactic similarity technique is computed over labels characterizing the couples of entities to be compared. For this, we have used a similarity based Edit-distance which consists on comparing two strings and computing the number of required edits (insertions, deletions and substitutions) of characters to transform one word into another. The syntactic similarity equation of two concepts c_1, c_2 is shown in (6), where $\mathrm{ed}(c_1, c_2)$ is the Edit-distance:

$$\mathrm{sim_{syn}}(\mathrm{c_1}, \mathrm{c_2}) = \frac{1}{1+\mathrm{ed}(\mathrm{c_1}, \mathrm{c_2})} \quad (6)$$

This structural similarity measure relies on the intuition that the elements of two distinct models are similar when their adjacent elements are similar. It is necessary to check if the concept under consideration is surrounded (descendants and generalizing) by similar concepts in the target ontology.

$$\mathrm{sim_{Struc}}(\mathrm{c_1}, \mathrm{c_2}) = \frac{(\mathrm{Sc}(\mathrm{c_1}, O1) \cap \mathrm{Sc}(\mathrm{c_2}, O2))}{|\mathrm{Sc}(\mathrm{c_1}, O1) \cup \mathrm{Sc}(\mathrm{c_2}, O2)|} \quad (7)$$

Where $\mathrm{Sc}(\mathrm{c_1}, O1)$ denotes the descendants and generalizing of the concept $\mathrm{c_1}$ in the ontology $\mathrm{O_1}$, and $\mathrm{Sc}(\mathrm{c_2}, O2)$ refers to the descendants and generalizing of the concept $\mathrm{c_2}$ in the ontology $\mathrm{O_2}$.

Finally the two kinds of similarity techniques between cluster's entities computed above are aggregated to determine the global similarity value.

$$\mathrm{sim_{Global}}(\mathrm{c_1}, \mathrm{c_2}) = 1/2(\mathrm{sim_{Struc}}(\mathrm{c_1}, \mathrm{c_2}) + \mathrm{sim}_{syn}(\mathrm{c_1}, \mathrm{c_2})) \quad (8)$$

5 EXPERIMENTAL RESULTS

In this section, we present some initial experimental results in order to evaluate the performance of the proposed method. We conduct a set of experiments applied on real world mammographic ontologies.

-'Breast Cancer Grading Ontology (BCGO)' (Bulzan, s.d.): The BCGO ontology has been developed in 2009; it contains 541 classes, 56 properties and 164 individuals. It is designed to be application oriented ontology and addresses the problem of semantic gap between high-level semantic concepts and the characteristics of the low-level image.

-'Mammo ontology' (Toujilov, 2012): The Gimi mammography ontology has been developed in 2012; it contains 692 classes and 135 properties, it is used to describe the richness and complexity of the domain and has been implemented with OWL 2, where the goal is to be integrated into a learning tool to compare the reviews of trainees with the expert annotations.

First, we proceed to compare the semantic distance with respect of an existing one called the structural proximity proposed in (Hu et al., 2008) and has been extensively used for ontology clustering such as (Ngo, 2012) and (Tu et al., 2005,) which is:

$$\mathrm{prox}(\mathrm{c_i}, \mathrm{c_j}) = \frac{2 * \mathrm{depth}(\mathrm{c_{ij}})}{\mathrm{depth}(\mathrm{c_i}) + \mathrm{depth}(\mathrm{c_j})} \quad (9)$$

Where $\mathrm{c_{ij}}$ is the common superclass of $\mathrm{c_i}$ and $\mathrm{c_j}$, and depth $\mathrm{c_i}$ gets the depth of $\mathrm{c_i}$ in the original class hierarchy.

For the clustering evaluation, we have used the cluster validity measures: Partition coefficient (PC) and Partition Entropy (PE). The PC indicates the average relative total of membership sharing among pairs of fuzzy subsets (Wanga and Zhang, 2007), where a high PC score designates a better partitioning. PC is computed as well:

$$\mathrm{PC} = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{c} {\mu_{ij}}^2 \quad (10)$$

The PE reveals the repartition of entities within the clusters (Jafar and Sivakumar, 2014), where a low score of PE indicates a better quality of partitioning.

$$\mathrm{PE} = -\frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{c} [\mu_{ij}\ log_2 \mu_{ij}] \quad (11)$$

The algorithm is implemented using Java language, with setting parameters as well: m = 2 and the number of clusters (not the same for all clusters). The algorithm converges when the centroids become stable. The histograms drawn in Figure 2 present the evaluation results of both distance metrics, where we notice that the algorithm reported good results for the relational context distance; where it generates for each data set maximum PC and minimum PE. We notice that, by using the structural proximity based

distance classes with weak depth tend to have low membership to different classes. Moreover, we find that, in most cases, medoids designate the classes with increased depths, which may lead to insignificant representative data, or the latter have to be as representative and general as possible among data in a cluster.

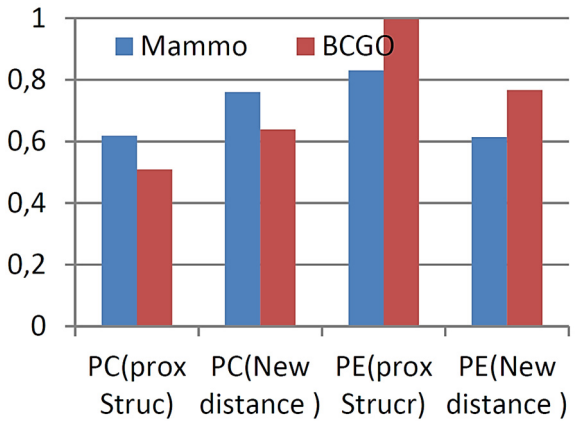


Figure 2: Evaluation of the proposed semantic distance.

To evaluate the alignment quality, we make a comparison between the alignments generated from our method and the ones generated from FALCON-AO (Ningsheng et al., 2005) and S-Match (Giunchiglia et al., s.d.). These systems are open source and available on the net. To this end, we adopt 3 standard known metrics widely used in data mining field: Precision, Recall and F-measure. We assume that M designates the set of correspondences discovered between ontological entities by the proposed tool. R is the set of reference correspondences found by the domain expert. These metrics are defined as follows:

- *Precision*: which represents the proportion of true positives among all matching elements found by the method. This allows qualifying the relevance of the alignment method: $P= |M \cap R|/ |M|$

- *Recall*: indicates the proportion of true positives among all matching elements in the reference alignment. This measure quantifies the cover of the alignment method: $R= |M \cap R|/|R|$

- *Fmeasure*: represents the harmonic mean between precision and recall. It compares the performance of methods by means of single measure: $F\text{-measure}=2.P.R/ (P+R)$

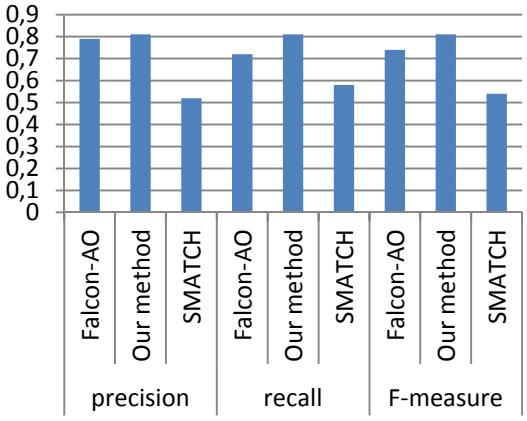


Figure 3: Alignment methods comparaison.

The Falcon-AO is a method that is based on partitioning the ontologies into crisp clusters before aligning the blocks. As regarding the S-Match tool, it is based on non-partitioning strategy; but it uses structural as well as element-based similarity techniques for correspondences discovering. The results in Figure 3 indicate that our fuzzy clustering-based method achieves a slight improvement in alignment quality as compared to the other existing tools. The reduced search space performs good precision by reducing the total of false positives number. Although the Falcon-AO system adopts ontology partitioning technique to reduce the complexity of the alignment problem, the proposed method is more efficient. This is due to benefit of the use of fuzzy clustering which increases the chance of finding correct alignments. As first observation, the use of fuzzy clustering has positively influenced the alignment quality. This confirms that:

-The use of clustering technique may reduce noticeably the scalability problem by reducing the search space.

-Assigning a concept to several clusters simultaneously increases the chance of discovering more correct alignments.

6 CONCLUSION AND FUTURE WORK

In this paper, we propose a fuzzy clustering alignment method. The main contributions of this paper are as follows:

-We present a fuzzy clustering method which consists on partitioning the ontology into fuzzy clusters where a concept may belong to several clusters simultaneously. To this end, we have

proposed a new semantic distance for clustering analyze.

-We introduce an approach to aligning clusters based on the predefined medoids. The latter may facilitate the knowledge base visualization, as well as speed up the task of matched clusters pairs.

-We proceed to align similar clusters' entities with the use of multiple similarity techniques.

As the next step, we are planning to ameliorate the system efficiency in terms of precision and recall, we are looking as well to perform experiments over large ontologies so to be able to participate in benchmark OAEI.

REFERENCES

Algergawy, S. Massmann & E. Rahm, 2011. A Clustering-Based Approach For Large-Scale Ontology Matching. Advances In *Databases And Information Systems*, January.Pp. 415-428.

Bulzan, S.D. *Bioportal.* [En Ligne] Available At: Http://Bioportal.Bioontolgy.Org/Ontologies/Bcgo [Accès Le 26 June 2009].

Bulzan, *Bioportal.* [En Ligne] Available At: Http://Bioportal.Bioontology.Org/Ontologies/Bcgo [Accès Le 26 June 2009].

Duan, Fokoue, A., K.Srinivas & B.Byrne, 2011. A Clustering-Based Approach To Ontology Alignment. *The Semantic Web–Iswc Springer,* Pp. 146-161.

Fernández, J.Velasco, I.J.Marsa-Maestre & M.Lopez-Carmona, 2012. Fuzzyalign: A Fuzzy Method For Ontology Alignment.. *Keod 2012 – Proceedings Of The International Conference On Knowledge Engineering And Ontology Development,* Pp. 98-107.

Giunchiglia, Autayeu, A. & Pane, J., S.D. S-Match: An Open Source Framework For Matching Lightweight Ontologies.. *Semantic Web,* 3(3), Pp. 307-317.

Hamdi, F. & Safar, B., 2009. Partitionnement D'ontologies Pour Le Passage A L'échelle Des Techniques D'alignement. *9eme Journées Francophones Extraction Et Gestion Des Connaissances.*

Hu, W., Qu, Y. & Cheng, G., 2008. Matching Large Ontologies: A Divide-And-Conquerapproach. *Data And Knowledge Engineering,* Volume 67, Pp. 140-160.

Hu, W., Zhao, Y. & Y.Qu, 2006. Partition-Based Block Matching Of Large Class Hierarchies. *Proceedings Of The First Asian Conference On The Semantic Web,* P. 72–83.

Idoudi, R., Ettabaa, K. S., Hamrouni, K. & Solaiman, B., 2014. An Evidence Based Approach For Multipe Similarity Measures Combining For Ontology Aligning. *1st Ieee International Conference On Image Processing Applications And Systems Conference (Ipas),* November.

Jafar, O. M. & Sivakumar, R., 2014. Hybrid Fuzzy Data Clustering Algorithm Using Different Distance Metrics: A Comparative Study. *International Journal Of Soft Computing And Engineering (Ijsce),* January, 3(6), Pp. 241-248.

Massmann, S. Et Al., 2011. Evolution Of The Coma Match System. *Ontology Matching,* June.Volume 49.

Ngo, D., 2012. *Enhancing Ontology Matching By Using Machine Learning, Graph Matching And Information Retrieval Techniques,* Montpellier: Université Montpellier Ii.

Ningsheng, Cheng, W. & Q.Yuzhong, 2005. Falcon-Ao: Aligning Ontologies With Falcon. *K-Cap Workshop On Integrating Ontologies,* Pp. 85-91.

Qiu & Liu, Y., 2014. An Effective Approach To Fuzzy Ontologies Alignment. *International Journal Of Database Theory And Application,* 7(3), Pp. 73-82.

Schlicht, A. & Stuckenschmidt, H., 2008. A Flexible Partitioning Tool For Large Ontologies. *Ieee/Wic/Acm International Conference On Web Intelligence, Wi,* December.P. 482–488..

Seddiquia, M. & Aono, M., 2009. An Efficient And Scalable Algorithm For Segmented Alignment Of Ontologies Of Arbitrary Size. *Web Semantics,* 7(4), Pp. 344-356.

Shvaiko & Euzenat, J., 2005. Survey Of Schema-Based Matching Approaches. *Journal On Data Semantics Iv,* Pp. 146-171.

Toujilov, P., 2012. Mammographic Knowledge Representation In Description Logic. *Springer,* August .Pp. 158-169.

Tu, K. Et Al., 2005,. Towards Imaging Large-Scale Ontologies For Quick Understanding And Analysis. *Proceedings Of The 4th International Semantic Web Conference, Lncs,* Volume 3729, P. 702–715.

Wanga & Zhang, 2007. On Fuzzy Cluster Validity Indices. *Fuzzy Sets And Systems,* 14 March, 158(19), P. 2095–2117.

Wang, Zhou & B.Xu, 2011. Matching Large Ontologies Based On Reduction Anchors. *Proceedings Of The Twenty-Second International Joint Conference On,* Volume 3, P. 2343–2348.

AUTHOR INDEX

Proceedings of ICEIS 2016 | VOLUME 1

18th International Conference on Enterprise Information Systems

WWW.ICEIS.ORG

Technically Co-sponsored:

IEEE SMC Systems, Man, and Cybernetics Society

In Cooperation with:

acm In-Cooperation

acm sigmis SIGCHI acm SIGAI

IRC Informatics Research Centre EIC SWIM IEICE Special Interest Group on Software Enterprise Modelling

ISBN:978-989-758-187-8