MLED_BI: a new BI Design Approach to Support Multilingualism in Business Intelligence

Nedim Dedić [1], Clare Stanier [1]

[1]Staffordshire University, ST4 2DE, Stoke-on-Trent, United Kingdom

Abstract **– Existing approaches to support Multilingualism (ML) in Business Intelligence (BI) create problems for business users, present a number of challenges from the technical perspective, and lead to issues with logical dependence in the star schema. In this paper, we propose MLED_BI (Multilingual Enabled Design for Business Intelligence), a novel BI design approach to support the application of ML in BI Environment, which overcomes the issues and problems found with existing approaches. The approach is based on a revision of the data warehouse dimensional modelling approach and treats the Star Schema as a higher level entity. This paper describes MLED_BI and the validation and evaluation approach used.**

Keywords **– Multilingualism, Data Warehouse, Business Intelligence, Data Mart Implementation, BI Design Approach.**

1. Introduction

Our previous work on ML in BI revealed that existing approaches to enable and support ML in BI are primarily ad-hoc workarounds, which lack a theoretical basis, or are vendor specific. These workarounds create problems for business users, present a number of technical challenges and lead to issues of logical dependence in the star schema [10]. We previously presented an approach to BI design that treats the Star schema as a higher level entity and saves textual descriptions from attributes and hierarchies elsewhere as language files. The proposal was supported by partial proof-of-concept (PoC) artefact developed to investigate the technical feasibility of the newly proposed approach [10]. In this paper, we extend the proposal and present the results of the full implementation and evaluation of the MLED_BI system. To support the evaluation of the MLED_BI approach, the implementation included the integration of a Multilingual Content Management System (MCMS) to enable content manipulation at presentational level in BI.

The reminder of the paper is organized as follows: section 2 discusses the background and motivation for the development of the MLED_BI design approach, section 3 discusses multilingualism and multilingual issues in BI environment. In section 4, we present the concept and architecture of MLED_BI in the context of BI/DWH design strategies, we explain the new approach to the star schema, and the role of the MCMS, and the implementation and testing approach used. Section 5 discusses the evaluation of MLED_BI from a technical and a user satisfaction perspective. Section 6 present the conclusions and recommendations for future work.

2. Background and Motivation

With emerging markets and expanding international cooperation, there is a requirement to support Business Intelligence (BI) applications in languages other than English, a process referred to as Multilingualism (ML) [10]. Some European countries such as Belgium [38], or Switzerland have several official languages meaning that ML may be a legal requirement. From a data quality (DQ) perspective, the quality requirements of interpretability and ease of use indicate that information should be made available to users in formats and languages which they can interpret [37].

DOI: 10.18421/TEM64-17
https://dx.doi.org/10.18421/TEM64-17

Corresponding author: Nedim Dedić,
Staffordshire University, ST4 2DE, Stoke-on-Trent, United Kingdom
Email: nedim.dedi@research.staffs.ac.uk

Received: 18 August 2017
Accepted: 12 October 2017
Published: 27 November 2017

This paper focuses on ML in the Data Warehousing (DWH) and Reporting layers of BI.

The motivation for this research developed from the design and performance problems encountered when implementing ML in a real world commercial BI environment. It was identified empirically that existing approaches to supporting multilingualism in a BI context created problems for business users, for example, slower information retrieval, delays in updating reports and difficulties in complying with legal requirements to provide data in more than one language. Additional problems include logical interdependence and coupling, such as the possibility of supporting only those languages at the reporting layer which are available in source systems, and the complexity of the processes required to change erroneous descriptive content in existing BI reports.

At a technical level, support for ML in BI presents a number of challenges including the additional complexity of the Extract-Transform-Load (ETL) processes required to support ML, excessive resource consumption, content dependency between systems, and data and processes redundancy. Some direct examples of these problems include redundancy of descriptive information stored in dimensional tables and the requirement to iterate the entire ETL process to support small changes in descriptive content.

The Star schema is identified as the DW dimensional modelling concept most used by industry [34], is recommended as the most appropriate design strategy for dimensional modelling and the development of data marts [19], [22], [26], and is very widely used in BI [17], [19], [22], [26], [28], [29], [34]. The MLED_BI approach presented in this paper is based around the Star schema.

Our previous work on ML in BI revealed that existing approaches to enable and support ML in BI are primarily ad-hoc workarounds, which lack a fully explained basis in the DW design literature, or are vendor specific [10]. The three most widely used approaches enable ML through the DWH layer and are based on the Star schema.

1) The additional attributes (AA) approach [18, 21]. The AA approach is derived from Kimball's method for delivering country-specific calendars [21], and recommends that where there are new language values for attributes, new attributes (fields) are added to the dimensional tables. This increases data volumes in the data warehouse,
2) The language identifier field (LIF) approach [18]. This extends the primary key to include a language identifier in dimension tables. However, this approach duplicates the number of the records in dimensions with every new language.
3) The additional dimensional tables/schemas (ADT) approach [7],[20],[21]. This implements as many dimensional tables as there are languages required. Different languages are saved in different database schema and/or in different tables.

As identified in previous work [10], the three approaches outlined above introduce redundancy and performance issues and a more efficient solution to the problem of ML in BI and DW is required. As discussed in section one, we previously presented an approach to BI design in which the Star schema is treated as a higher level entity; textual descriptions from attributes and hierarchies are saved elsewhere as language files. The proposal included a ML Content Management System (MCSM) to enable content manipulation at the reporting layer based on the modified Star schema and was supported by a proof-of-concept (PoC) artefact developed to investigate the technical feasibility of the approach [10]. In this paper, we extend the proposal and present the results of the full implementation and evaluation of the MLED_BI system.

3. Multilingual Issues in BI

Multilingualism is an individual and social phenomena that requires the acquisition, knowledge and use of several languages by communities or individuals, and usually implies two or more languages [5]. In the context of this paper, Multilingualism in Business Intelligence is seen as the ability to store descriptive information at data warehousing level and to use this information at presentation level in the form of reports, queries or dashboards in more than one language [10]. ML in BI is the term used to describe the process of providing descriptive content in BI reports in more than one language. The complexity of ML in BI is visible from Figure 1.

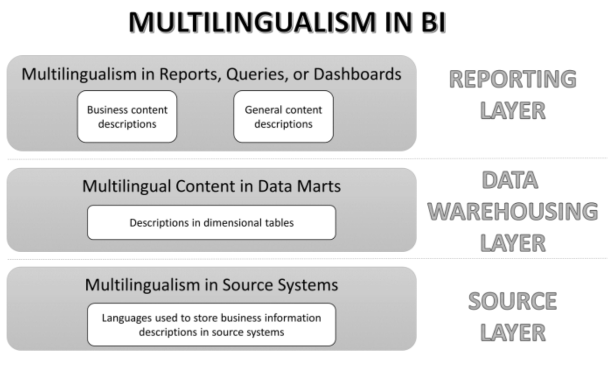

Figure 1. The full complexity of ML in BI

At the BI **source layer**, ML encompasses the concept of languages used to store business information descriptions from operational systems conventionally known as master data [24, 32, 35]. In this research, the terms business information descriptions and master data are used interchangeably as they represent the same concept. Master data are used to describe the entities, which are independent of and fundamental to the enterprise operations and because they describe things that are critical to the organization's operations, such as products, persons, customers, locations, suppliers, or services, they are sometimes seen as "nouns" [35]. The purpose of master data is to describe, categorize, aggregate, or evaluate transactional data, while transactional data describes the activities and transactions of the business, and are generated by the operational system [32]. Thus, transactional data are created during business processes, such as placing an order by customer, or purchase by supplier, while master data are independent of specific orders [24]. The multilingual context of BI applies to the use of master data, making transactional data out of the scope of this research.

At the **DWH layer** ML is concerned with the dimensional modelling of business information descriptions (master data) and storage of these descriptions in dimensional tables at data warehouse (DW) or data mart (DM) level.

At the **reporting layer** there are two types of multilingual content a) business information descriptions (master data), and b) general content/report descriptions. The focus of this paper is business information descriptions (master data). Business information descriptions at the reporting layer are the same as the business information descriptions used at the source layer level and the business information descriptions saved in dimensional tables at DWH layer. General content/report descriptions are used to provide general information about a report, query or dashboard, such as title or other similar information, and to provide descriptions for activity-based content, such as menus or filters.

4. MLED_BI: A New BI Design Approach to Support ML in BI

As discussed in section 2, the Star schema concept provided the underpinning basis for the development of MLED_BI, but MLED_BI was also influenced by the dynamic content concept for multilingual websites patented by Kumhyr [23]. Kumhyr proposed the use of content strings identified by content keys with values retrieved from a data store based on language preference. Setting language preference could be achieved, for example by using HTTP Methods, such as GET and POST. Referencing our initial findings, Kumhyr's concept, and the idea of using separate HTML language files to overcome issues of ML in web as proposed by Lepouras & Vassiliakis [25], the new BI design approach, MLED_BI, was developed. The new BI design approach treats the star schema as a higher level entity and proposes saving textual descriptions from attributes and hierarchies elsewhere as language files – outside of dimensional tables. This provides not only physical, but logical data independence.

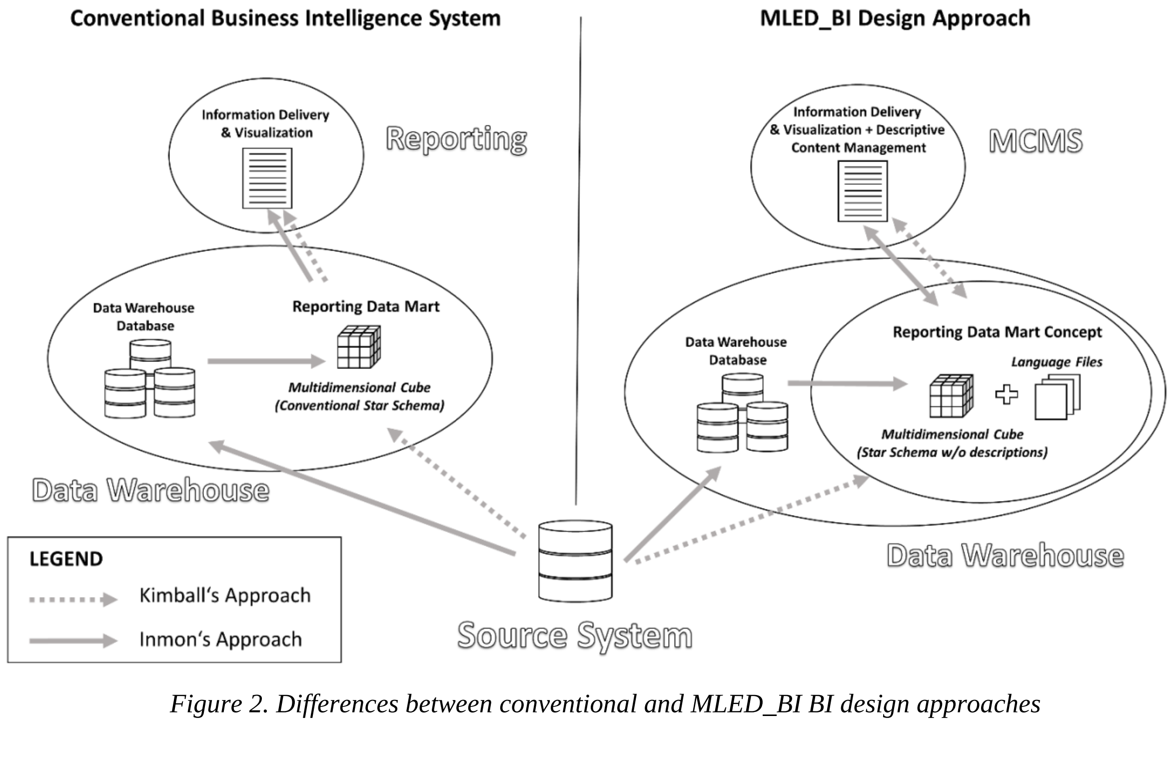

Figure 2. Differences between conventional and MLED_BI BI design approaches

This approach to data mart implementation also provided the possibility of incorporating the Content Management System concept into MLED_BI to support direct management of descriptive content in BI by wider groups of stakeholders. The next section explains existing BI design approaches and compares them to MLED_BI.

4.1. BI/DW Design Strategies

There are two main design approaches for the development of BI/DW, one associated with Inmon and one with Kimball. A third approach, the Data Vault approach [26], supports the Inmon design concept but has some differences in DM architecture. As can be seen from Figure 2, MLED_BI can be used with both the Inmon and Kimball data warehouse design approaches. This allows the MLED_BI approach to be integrated into existing BI/DW environments, whether based on the Inmon or Kimball philosophy without making any major changes to existing system architecture. For example, from the staging area, data can be extracted directly into reporting data marts (dashed arrow in Figure 2), following the philosophy of Kimball, or to the DW and then to reporting data marts, following the Data Vault and Inmon philosophies (full arrow in Figure 2). This part of our approach largely follows existing DW design strategies.

4.2. New Approach to Star Schema

Previous work identified that the main challenge to providing support for ML in BI in the context of the Star schema is that attribute and hierarchy descriptions are saved inside the dimensional tables of data marts [10]. This leads, as previously discussed, to performance problems and problems of dependency and coupling. A snowflake design approach reduces redundancy but is highly normalised which introduces other issues [10].

The alternative approach proposed here is that before data is stored in reporting data marts, descriptive information is extracted (master data descriptions, such as attributes and hierarchical descriptions) together with their IDs to language files stored elsewhere, on the server for example. As attributes and hierarchical descriptions and their IDs are extracted to separate language files, only integer values (descriptions IDs) are stored in dimensional tables. Descriptions of attributes and hierarchies are 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 avoids redundancy, description-based aggregations and source system language dependency [10]. Figure 3 provides an overview of data marts based on existing approaches while Figure 4 provides an overview based on the MLED_BI approach.

As shown, a data mart based on the existing Star schema approaches consists of a Fact table holding transactional data and foreign keys to dimensions and Dimensional tables holding descriptive master data. A Star schema based on MLED_BI approach, as shown in Figure 4, consists of a Fact table holding transactional data and foreign keys to dimensions; dimensional tables hold only master data IDs and link to language files with arrays holding descriptive information.

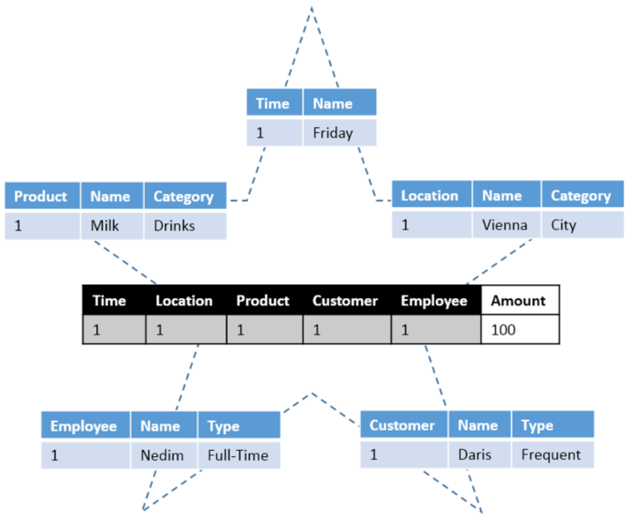

Figure 3. DM based on conventional Star schema

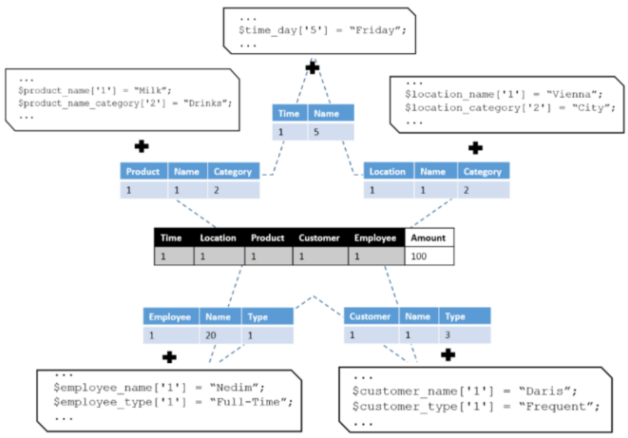

Figure 4. DM based on MLED_BI approach

4.3. Multilingual Content Management System (MCMS)

The use of MCMS, which is related to the Reporting layer, is the second major difference between existing BI/DWH design approaches and MLED_BI. Multilingual content management is essential for the availability of information in local languages [1]. If there is a need for multiple languages, then there is an imperative to enable transfer and processing of textual values for localization purposes [36]. A simple solution, proposed by Dempsey [14], was to copy existing HTML pages for new languages and change HTML paragraph values according to the new language requirement, but this does not have sufficient functionality for a dynamic semantic web management or for a complex BI system. A Multilingual CMS (MCMS), would help to overcome the technical challenges of multilingual content [1].

Existing BI/DWH design approaches support only the following activities in reporting layer: viewing of reports and associated activities such as drill up/down, selection, filtering, other analytical operations and browsing; re-execution, sharing and changing languages. Content changes are permitted only in the source system and changes are not made at DW level. For this reason the BI reporting layer provides only visualization of the data stored in the DW. This restriction on content change excludes additional data generated directly in DW itself through transformations and operations on existing data. The Star schema uses attribute descriptions and hierarchies as a basis for data aggregation and representation in reports, thus, this “no data change philosophy” is reasonable.

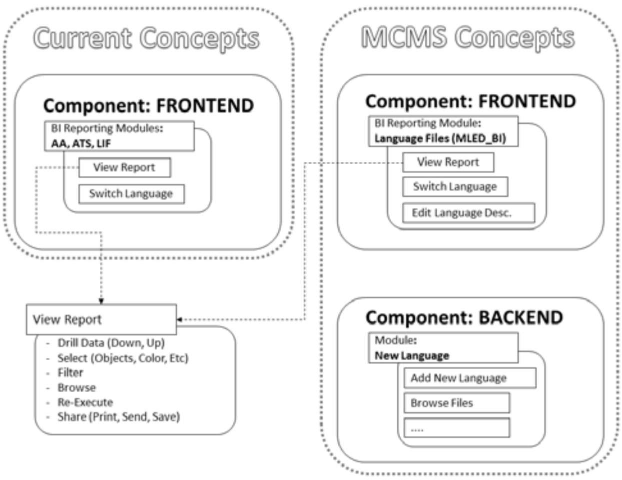

Figure 5. Conventional reporting layer and one that includes MLCSM

Using the MLED_BI approach, Star schema do not use attribute descriptions and hierarchies as the basis for a data aggregation, but operate only with the IDs. Thus, it is possible to change descriptions without creating unrelated data or additional categories for the same data at reporting level or in the data warehouse. Using the MLED_BI MCMS approach, activities at the reporting level can include the viewing, execution, sharing and switching language operations supported by existing systems and can provide further functionality such as editing descriptions for existing languages directly via a web interface and adding new languages and their variations directly by business users, independently of the existing languages in source systems. The difference between existing implementations of the BI reporting layer and the MCMS approach are shown in Figure 5.

In the MCMS, there are two aspects to the web interface. The frontend element enables execution of the standard activities found in conventional BI reporting approaches. The backend provides additional functionality, allowing editing and management of content and inclusion of additional languages. Backend functionality could be extended with additional modules, for example to enable the execution of ETL processes by business users or to edit various aspects of web interface.

The MLCSM approach allows business users to change erroneous descriptive content directly; this would simplify or possibly in some chases eliminate, the ETL processes required to perform language changes. As discussed in section 2, in existing approaches, the whole ETL process needs to be re-executed to change a descriptive value in a BI report for specific master data. It is, however, important to note that it is highly recommended that values in the source system should also be updated to reflect the corrections made in language files. This process can be automated for business users by implementing a trigger to notify the relevant department of the change made, so that the source system can then be updated. The justification for propagating changes to the source system is that if there were a future need to load the whole master data for specific dimension, this would overwrite corrections made. It is recommended that language files are amended through the MCMS for small language corrections and that standard ETL processes are executed when dealing with a larger number of corrections at the same time.

4.4. Testing

To test the proposed design approach, MLED_BI, four different ML BI systems were built, each using the same data but designed according to a different approach. One system used the AA (additional attributes in dimensional tables) approach, the second system used the LIF (language identifier field in dimensional tables) approach, the third system used the ATS (additional schema/tables for dimensions) and the fourth used MLED_BI (based on language files + MCMS). The rationale for developing the different systems was to enable objective comparison of measurements and to facilitate obtaining user opinions. The implementation comprised of three main layers:

1) **Source Layer** in the form of Sample Source System Database (SSSD). The same SSSD was used for every BI system implementation.
2) **Data Warehousing Layer**. Four different dimensional modelling methods to enable ML in BI were used. All were based on the Star schema.
3) **Reporting Layer**. Every approach had an element at the Reporting Layer. This included three reports based on conventional BI design approaches reflecting the three different methods of data mart implementation (AA, LIF, and ATS) plus an MCMS reflecting the MLED_BI approach of data mart implementation based on Language files extension, which also included appropriate BI report.

5. Evaluation

The evaluation was based on an Evaluation tool we developed previously, which measures the success of amendments or updates to existing BI solutions to support improved BI reporting [11], [39]. The development of this tool elicited two clusters of measurements: technical functionality and business/end user satisfaction.

5.1. Metrics Based Evaluation

Technical functionality was identified as one of the clusters of measurements to be considered when measuring the success of changes to BI reporting systems. Eleven technical measurements were identified [11, 39], which are presented in Table 1, and are used in this paper to evaluate the technical effectiveness of MLED_BI. The measures covered elements such as speed of execution and memory consumption and are labelled TM1 through to TM11.

Table 1. Technical Functionality Measurements [11, 39]

Code	Measurments
TM1	*- Speed of execution time for Initial BI report or dashboard*
TM2	*- Speed of execution time for SQL query*
TM3	*- Speed of re-execution time when changing report language, currency or unit*
TM4	*- Speed of execution time when drilling-down, conditioning, removing or adding columns in reports*
TM5	*- CPU memory usage during execution of initial BI report or dashboard*
TM6	*- CPU memory usage during re-execution of report when changing language, currency or unit*
TM7	*- CPU memory usage during execution of SQL query*
TM8	*- Database memory consumption*
TM9	*- Amount of Time required to change erroneous descriptions of descriptive attributes and hierarchies*
TM10	*- Technical scalability of proposed solution in the existing environment*
TM11	*- Support for possible extension of the system in the future*

Every BI report used for testing included code that measured and provided information about the execution speed of the whole web application and the relevant SQL query. The fact table holding transactional data in any underlying DM had 1.199.989 records; the number of records in the dimensional tables reflected the requirements of the respective implementation method. Despite using different structures based on different DM implementation methods, the dimension tables were implemented to support providing the same data to the end user via BI reports.

TM1 "Speed of execution time for initial BI report or dashboard" and TM2 "Speed of execution time for SQL Query" were identified as required technical measurements (Table 1). To support measurement and evaluation, each BI report for each implementation method was executed 20 times in the same environment and provided the same data to the end user.

Speed execution data are shown in Table 2. Based on the recorded values in Table 2, both TM1, "Speed of execution time for initial BI report or dashboard" and TM2 "Speed of execution time for SQL Query", showed improved speed performance when using BI reports supported by a data mart based on the FILES implementation method, which is a part of the MLED_BI concept.

Table 2. Execution speed for initial BI report and underlying SQL Queries

Initial Report Execution								
	AA		ATS		LIF		MLED_BI	
Attempt	Complete Web Report	SQL stat. used in Web Report	Complete Web Report	SQL stat. used in Web Report	Complete Web Report	SQL stat. used in Web Report	Complete Web Report	SQL stat. used in Web Report
1	12,5275	12,5233	12,4374	12,4344	16,2937	16,2902	7,4950	7,4898
2	12,4332	12,4298	12,4122	12,4090	16,0858	16,0823	7,5801	7,5749
3	12,5809	12,5778	12,4608	12,4575	16,1425	16,1392	7,6185	7,6134
4	12,6224	12,6190	12,4000	12,3967	16,2740	16,2706	7,5277	7,5224
5	12,4993	12,4960	12,4355	12,4324	16,3258	16,3224	7,6781	7,6729
6	12,5360	12,5328	12,5702	12,5669	16,2008	16,1975	7,5741	7,5690
7	12,6437	12,6405	12,4289	12,4223	16,3136	16,3105	7,5398	7,5348
8	12,5897	12,5864	12,4190	12,4157	16,2899	16,2864	7,5564	7,5510
9	12,5156	12,5123	12,5165	12,5132	16,1993	16,1885	7,5491	7,5437
10	12,6172	12,6140	12,3446	12,3410	16,4247	16,4214	7,5749	7,5696
11	12,5074	12,5042	12,4005	12,3972	16,2205	16,2174	7,6166	7,6103
12	12,7594	12,7564	12,5111	12,5079	16,1771	16,1588	7,6618	7,6564
13	12,6050	12,6018	12,4352	12,4320	16,4911	16,4876	7,6142	7,6090
14	12,6377	12,6343	12,4342	12,4311	16,1792	16,1760	7,5595	7,5543
15	12,6069	12,6038	12,4079	12,4044	16,4089	16,4057	7,6616	7,6561
16	12,6464	12,6432	12,4766	12,4735	16,3596	16,3562	7,5892	7,5834
17	12,5294	12,5261	12,3616	12,3584	16,2835	16,2805	7,5122	7,5069
18	12,5773	12,5742	12,3611	12,3577	16,2369	16,2337	7,6045	7,5989
19	12,6389	12,6355	12,6371	12,6338	16,2732	16,2580	7,5885	7,5824
20	12,6754	12,6721	12,4772	12,4743	16,1471	16,1440	7,6106	7,6048
Average	**12,5875**	**12,5842**	**12,4464**	**12,4430**	**16,2664**	**16,2613**	**7,5856**	**7,5802**

The next technical measurement identified as relevant was TM3 "Speed of re-execution time when changing report language, currency or unit". Multilingual issues in BI, especially those related to the business content descriptions (master data), are the focus of this paper, thus the interest is only in measuring re-execution time when changing the reporting language. Changing currency or unit descriptions in BI reports reflects the issues involved in changing the reporting language for any other business content. Currency or unit recalculations or transformations on transactional data are not relevant for business information descriptions (master data) and are not considered in this research.

To evaluate TM3, “speed of re-execution time when changing report language”, the report language was changed 20 times in a previously executed BI report in the same environment. The same master and transactional data was used throughout. The results showed that the MLED_BI approach provides a significant advantage (Table 3).

Comparison between Table 2 and Table 3 shows that changing report language for a BI report based on conventional BI design approaches requires as much time as initial report execution. This is due to fact that in the conventional BI design approach the SQL query must be re-executed to provide business content descriptions in other language. However, this is not the case in the MLED_BI design approach. As shown in Table 3, the time required to change the preview language of an already executed BI report was less than a hundredth of a second. The reason for this improvement is the fact that there is no need to re-execute SQL query, as the new language file was loaded and applied to an already existing SQL result set. Because of that, there are no SQL execution times recorded in Table 3.

Table 3. Execution speed for language change in already executed reports

Language Switch in Already Executed Report								
	AA		ATS		LIF		MLED_BI	
Attempt	Complete Web Report	SQL stat. used in Web Report	Complete Web Report	SQL stat. used in Web Report	Complete Web Report	SQL stat. used in Web Report	Complete Web Report	SQL stat. used in Web Report
1	12,6434	12,6399	12,3644	12,3609	16,3424	16,3390	0,0018	0,0000
2	12,5360	12,5326	12,5002	12,5468	16,5138	16,5099	0,0020	0,0000
3	12,6559	12,6526	12,4133	12,4099	16,3648	16,3615	0,0019	0,0000
4	12,6109	12,6058	12,3949	12,3915	16,2968	16,2938	0,0017	0,0000
5	12,6787	12,6757	12,6204	12,6172	16,4638	16,4607	0,0028	0,0000
6	12,5642	12,5608	12,3574	12,3541	16,2970	16,2939	0,0019	0,0000
7	12,6296	12,6264	12,5167	12,5137	16,2809	16,2775	0,0017	0,0000
8	12,6848	12,6814	12,5724	12,5691	16,2914	16,2879	0,0024	0,0000
9	12,7347	12,7313	12,4189	12,4160	16,3695	16,3664	0,0017	0,0000
10	12,5303	12,5269	12,5835	12,5802	16,2624	16,2591	0,0024	0,0000
11	12,5255	12,5221	12,4638	12,4606	16,4954	16,4924	0,0019	0,0000
12	12,7773	12,7740	12,4307	12,4275	16,2686	16,2653	0,0017	0,0000
13	12,5921	12,5746	12,5570	12,5533	16,2960	16,2925	0,0017	0,0000
14	12,6057	12,6025	12,5353	12,5418	16,4996	16,4962	0,0018	0,0000
15	12,5757	12,5727	12,4645	12,4613	16,3162	16,3129	0,0027	0,0000
16	12,6254	12,6219	12,4685	12,4654	16,2748	16,2714	0,0017	0,0000
17	12,5687	12,5654	12,5770	12,5738	16,3962	16,3928	0,0018	0,0000
18	12,7437	12,7406	12,4703	12,4669	16,3274	16,3240	0,0018	0,0000
19	12,5493	12,5456	12,4770	12,4740	16,4062	16,4024	0,0020	0,0000
20	12,5773	12,5741	12,4858	12,4826	16,5049	16,5012	0,0017	0,0000
Average	**12,6205**	**12,6163**	**12,4836**	**12,4833**	**16,3634**	**16,3600**	**0,0020**	**0,0000**

TM4 “Speed of execution time when drilling-down, conditioning, removing or adding columns in reports” was identified as a relevant technical measurement. However, performance of those processes reflects the performance of initial BI report execution. Although sometimes visually implemented as a function of an existing report, drilling-down, conditioning, removing or adding new columns is in fact nothing more than the execution of a new report under new criteria, or with different columns at different level of business content.

CPU memory usage during the execution of the initial BI report or dashboard, during execution of SQL query, and during re-execution of report when changing language, currency or unit are identified as relevant measuments. In that context, TM5 “CPU memory usage during execution of initial BI report or dashboard” was measured. CPU memory usage during the execution of the initial BI report or dashboard was monitored using the built-in functionality of phpMyAdmin which enables CPU status monitoring for any process executed on the localhost. During the execution of BI reports based on any method or approach, CPU system usage was between 20% and 40%. No significant differences are identified for any DM implementation method or for any BI design approach.

To measure and compare TM6 “CPU usage during re-execution of report when changing language”, a language changing process in a previously executed BI report was activated while the CPU status of the web application was simultaneously monitored. The same process was applied for each BI report developed for each system. Figure 6 shows that a language change process in a BI report based on the MLED_BI design approach offers more optimal resource usage than the same processes based on a conventional BI design approach. Changing the language in BI reports based on conventional DM implementation methods requires almost the same CPU resources as the initial BI report execution, which is somewhere between 20% and 40%. This behaviour was expected as the SQL query for the required language needs to be executed again, this time taking business information descriptions from the database in another language. However, this is not the case with BI reports implemented using MLED_BI. The language changing process for a BI report based on this approach had CPU usage of 10% or less. This is explained by the fact that there was no need to rerun the SQL query to acquire business information descriptions (master data descriptions) in other languages and the CPU was used only to load and apply another language file in the existing web application. This is useful in environments with limited CPU resources as it could enable smoother operations with BI reports for a larger number of users. It could also prevent problems that might be created by excessive use of CPU.

To measure the speed of execution of the web application or a part of that application, such as an SQL query, a modular approach can be used, for example, implementing measuring code at appropriate places would be sufficient. This approach was used to measure TM1 “Speed of execution time for initial BI report or dashboard” and TM2 “Speed of execution time for SQL Query”. CPU usage could also be measured by executing the whole web application. However, to measure and compare TM7 “CPU usage during execution of SQL query only”,

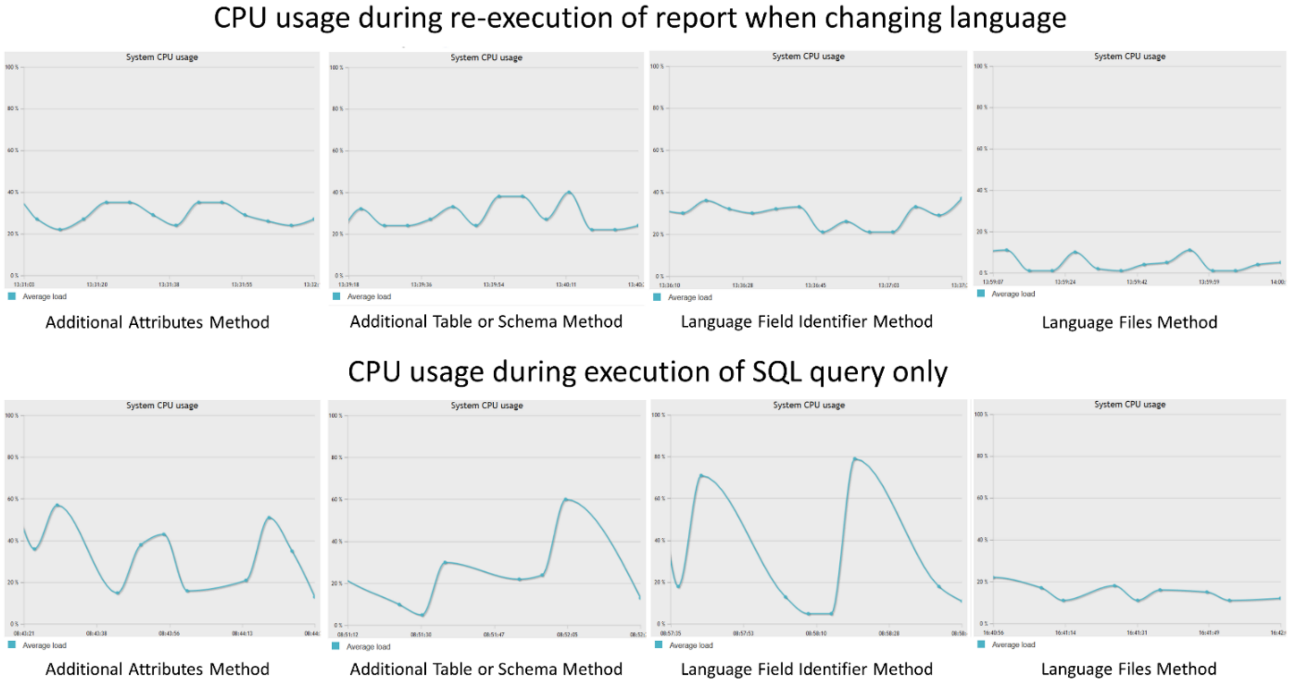

Figure 6. CPU usage comparison

an environment independent of previously developed BI reports or the web application was needed.

This is due to the fact that each BI report requires the execution of different code. The phpMyAdmin application environment was used for this element. It is important to note that using the phpMyAdmin environment itself requires additional CPU resources to enable the execution of SQL queries. However, this applied to all queries and the purpose of the test was to establish which approach had more optimal CPU usage rather than to establish the actual level of CPU usage for each item. As shown in Figure 6, a query on the DM based on MLED_BI implementation methods was observed to have the most optimal CPU usage. While other SQL queries had large oscillations in CPU usage rising as high 80%, this SQL query had linear usage of CPU resources barely exceeding 20%.

The next measure used was TM8 “Database memory consumption”. Differences in memory consumption are not identified as significant due to the small volume of master data. However, differences in memory consumption would be drastically changed if the sample product dimension had 4.000.000 records, which is currently standard Walmart product pallet [10]. It was anticipated that MLED_BI would reduce database memory consumption in the DM given that business information descriptions are stored outside the database as language files elsewhere on the server. Taking into account the cumulative requirement for memory to store information to the DM, including sever memory requirements for storage if language files, the actual advantage of the MLED_BI approach for this element is arguable.

TM9 “Amount of time required to change erroneous descriptions of descriptive attributes and hierarchies”: As there are significant structural differences between MLED_BI and the other ML design approaches, a standard measurement and comparison process was not appropriate. Error changing activities in BI reports based on conventional BI design approaches requires external human intervention and communication with other teams. This is not the case with the MLED_BI approach. For this element, validation was through use of an expert panel, composed of six BI domain experts, from three countries (Germany, Austria and Slovenia) and from three different companies. The domain experts had technical and user understanding of BI processes and had more than 50 years of combined BI experience.

In a simple BI report implemented using the MLED_BI approach, less than 30 seconds would be needed to change erroneous business information descriptions. Using the MLED_BI approach with a previously executed BI report, a business user can select an erroneous description. This action leads to a landing page in a web environment where the user is allowed to change the erroneous content directly in the relevant language file. There is no need to communicate with any other team or to wait for processes to be executed. In an ideal environment, for BI reports implemented as a part of BI system based on a conventional design approach, the process of changing business information descriptions would take a minimum of two hours. Industrial observation and discussions with the BI domain experts identified a timescale of between 24 and 36 hours as the standard timescale for the application of changes to business information descriptions in BI reports. During the evaluation, which also included semi-structured interviews, business users identified the delays in changing erroneous content as one of the most frustrating aspects of working with reports in BI systems based on conventional design approach. The MLED_BI approach offers a clear benefit in terms of speed and flexibility.

The remaining technical measurements identified as relevant when measuring the success of changes to support better BI reporting are TM10 "Technical scalability of proposed solution in the existing environment" and TM11 "Support for possible extension of the system in the future". These factors cannot be measured using metrics in the same way as, for example, CPU usage. Instead, domain experts were asked for their judgements as to whether the MLED_BI approach would be scalable and extensible. The use of separate language files means that additional languages can be added easily and without needing to amend the Star Schema. The decoupling of descriptions in specific languages from descriptive content in the Star Scheme itself, promotes logical independence, supporting extensibility. Based on the evaluation of the domain experts, MLED_BI was found to support scalability (TM10) and extensibility (TM11).

6.2. Scalability and Extensibility

A further issue was the integration of the MLED_BI approach in an existing BI environment. MLED_BI uses a modular approach and because it is based on the widely used Star Schema construct, it does not require a complete redesign of the existing system. MLED_BI can be applied as an additional module within an existing BI system or can be implemented as a new standalone BI system. Implementing MLED_BI in an existing BI system would require creation of language files, and adding additional columns to dimensional tables. Those columns would hold attribute IDs to reference existing attributes with language files. MLED_BI would also require extending existing ETL processes, and developing new BI reports as a part of the MCMS concept at reporting level. The creation of new data marts or new dimensional tables is not required and existing BI reports can be retained and used in parallel with new reports based on MLED_BI. Extending existing dimensions with additional columns does not require the deletion or modification of any data. Existing ETL processes can be extended to extract the required for reports based on the MLED_BI approach, but the data content of existing BI reports would not be affected. This allows the organisation to roll back to its previous approach if this is required for any reason. In addition, previous BI reports could be integrated into the MCSM. The MLED_BI BI design approach supports full integration with existing BI systems. Any existing workarounds to support ML in BI would require creation of new dimensional tables to be integrated with any other existing workaround. This is due to the fact, that every conventional DM implementation method that supports ML has a specific architecture of dimensional tables.

Consequently, creating new dimensional tables requires new ETL processes, new BI reports, and at the end, loading of the new data to support changes made. This is in effect a new implementation of the BI system. Moreover, once is a new BI system based on any existing DM implementation method using a conventional BI design approach had been created, it would be very hard to get back to the old system.

The MLED_BI approach supports the use of all languages available in the source system in BI reports. However, the number of languages used in BI reports is independent of the number of languages available in source systems. Subject to the necessary consideration of resources to transfer content, to enable additional languages for BI reports based on MLED_BI, it would be sufficient to provide only a language file with content for the new language. As soon as a new language file is available on the server, business users could use BI reports in that language. In the contrast to the existing BI design approaches, in the MLED_BI approach there is no need to implement and enable a language in all source systems, to modify ETL processes to support the new language, and to modify dimensional tables to support the new language. Industrial experience shows that this is highly beneficial where there is a need to support BI reporting in languages or dialects that are generally not available in source systems.

A limitation of the MLED_BI approach is that the initial design and implementation requires more resources for the design and development phases than conventional BI design approaches although producing benefits in terms of reduced processing and greater flexibility further down the data chain.

6.3. Business / user satisfaction

User satisfaction is regarded as a key measure in BI [6], [8], [9], [11], [12], [13], [30], [31] and the MLED_BI approach was evaluated for user satisfaction as well as technical effectiveness. Where participants have a high level of knowledge and expertise in relation to the research area, four to five participants are a sufficient sample size to achieve data saturation in qualitative interviews [33]. Guest, Bunce & Johnson propose a range of between 6-12 participants for projects having a narrow research scope focused on a homogenous target audience [16]. Miller sees a sample size of 6-70 as sufficient taking into account the scope of research and resources available [27]. Bonde [4] identified that most of the scientists propose a 1+ sample size according to the research scope and type of inquiry as sufficient sample size for data saturation; meaning that the appropriate number of respondents can be between one and any other number depending on scenario and complexity of research field [2],[3],[15].

To evaluate MLED_BI from a user perspective, six business users who identified themselves as key BI users, coming from three international companies using multilingual BI systems were interviewed. Interviews were held in three different countries (Austria, Slovenia and Croatia). The evaluation sessions were carried out as follows: a presentation given to the business user, explaining MLED_BI and the differences compared to conventional ML BI design approaches. The artefacts developed to validate MLED_BI, including the MCMS were demonstrated. The demonstration covered the three existing approaches to support ML in BI (AA, LIF, ADT) and the MLED_BI approach. The business users were able to experience the functionality and differences between the four approaches. This was followed by completion of an evaluation questionnaire, which was based on user satisfaction cluster of measurements extracted from evaluation tool developed in previous work [11, 39]. User satisfaction measurements are listed in Table 4.

Table 4. User satisfaction measurements [11, 39]

Code	***User Satisfaction***
BM1	*- Information content meets your needs?*
BM2	*- The information provided in the reports is accurate?*
BM3	*- Output is presented in a format that you find useful?*
BM4	*- The system and associated reports are easy for you to use?*
BM5	*- Information in the reports is up to date?*
BM6	*- Reports have the functionality that you require?*
BM7	*- The BI system is flexible enough to support easy change of "descriptive content"?*
BM8	*- Is the change of "descriptive content"* fast enough to fulfil business requirement?*
BM9	*- Exporting and sharing content functionalities meet your needs?*

As an introduction in evaluation questionnaire a following scenario was provided, and used a basis for the for validation from business users:

"As a business key user, you want to browse a Business Intelligence based report that provides products sales overview. This report should provide overview of sales per year, product area, category and subcategory and include gross sales, net sales and profit as appropriate metrics. All reports you visit as a part of this demonstration, should provide same data based on same source systems; however, their implementations are based on different design approach philosophy. The first three approaches have different design philosophies only in regard to data marts, while fourth applies different design approach to whole Business Intelligence concept.

Your task is to test every approach concerning "application of multilingualism" according to moderators instructions and give your opinion by filling in questionnaire and giving your feedback."

As all BI reports provided the same content and provided scenario assumes that information content in BI reports meet the needs of business users, the first question (BM1) from Table 4, namely "Information content meets your needs?" was not included into MLED_BI evaluation process.

All business users answered with "Yes" to all following questions for all presented BI reports based on any BI design approach or any DM implementation method: *(BM2) "The information provided in the reports is accurate?"*, *(BM3) "Output is presented in a format that you find useful?"*, *(BM4) "The system and associated reports are easy for you to use?"*, *(BM5) "Information in the reports is up to date?"*, and *(BM9) "Exporting and sharing content functionalities meet your needs?"*. Due to the nature of the scenario, the application of ML in BI, and the output of the BI reports presented in the demonstration this answers was expected. A conclusion would be that every BI design approach supported by any DM implementation method has the capability to provide BI reports that meet user needs and to provide a BI system that delivers accurate information presented in useful format, reports that are easy to use, are up to date, and have appropriate content sharing functionalities. In this context, we found no advantage of MLED_BI over existing BI design approaches, or DM implementation method.

However, based on the scenario, only BI reports developed on the MLED_BI design approach received "Yes" from all business users as an answer to the following questions: *(BM6) "Reports have the functionality that you require?"*, *(BM7) "The BI system is flexible enough to support easy change of "descriptive content"?"*, and *(BM8) "Is the change of descriptive content fast enough to fulfil business requirement?"*. This confirmed that one of the advantages of MLED_BI, comparted to conventional BI design approaches, is that the greater data independence supported by the MLED_BI approach, enables the user to carry out activities such as changing the language of already executed report, making corrections to erroneous content, or enabling new languages for reports. In additional discussion with the same business users, users reported satisfaction with the BI reports based on MLED_BI design approach.

6. Conclusion and Future Work

This paper presented MLED_BI, a new design approach which supports ML in BI environment. MLED_BI proposes a new understanding of the star schema as a higher level entity, enabling textual descriptions of master data to be stored outside dimension tables as language files. This in turn supports the integration of the Multilingual Content Management System into the BI system to enable multilingual content manipulation at presentational level in BI.

An evaluation of technical functionality showed the advantage of MLED_BI compared to conventional BI design approaches in terms of:

- *Speed of execution time for Initial BI report or dashboard;*
- *Speed of execution time for SQL query;*
- *Speed of re-execution time when changing report language, currency or unit;*
- *CPU memory usage during execution of initial BI report or dashboard;*
- *CPU memory usage during execution of SQL query;*
- *CPU memory usage during re-execution of report when changing language, currency or unit;*
- *Amount of Time required to change erroneous descriptions of descriptive attributes and hierarchies;*

The technical functionality measurement “Database memory consumption” also showed some advantage when using MLED_BI but as discussed in section 6.1, the benefits are arguable if other factors such as data volumes are taken into consideration. Evaluation with users indicated that MLED_BI is more scalable and more easily integrated into existing BI environments than conventional approaches.

An important limitation of the MLED_BI approach is that the initial design and implementation requires more resources for the design and development phases than conventional BI design approaches. For larger organisations, this initial increased resource demand would be outweighed by benefits, such as increased performance and flexibility in data management, following implementation. For smaller companies, however, and particularly those that do not operate in multilingual environment the benefits of MLED_BI would be questionable.

The evaluation of user satisfaction confirmed the benefits of MLED_BI, including the multilingual content management system, compared to conventional ML BI design approaches in respect of activities such as changing language of already executed report, making corrections to erroneous content, or enabling new languages for reports. However, no advantage was identified, compared to conventional approaches, in terms of provision of BI reports suggesting that for non-technical users, one of the main benefits of the MLED_BI approach is the greater flexibility and ease of data manipulation that MLED_BI provides.

Further work includes developing the MLED_BI approach further, applying it in a real world environment and evaluation of other types of multilingual content within MLED_BI, such as currencies, and time representations.

References

[1]. Arefin, M. S., Morimoto, Y., & Yasmin, A. (2011, April). Multilingual Content Management in Web Environment. In *Information Science and Applications (ICISA), 2011 International Conference on* (pp. 1-9). IEEE.

[2]. Back, L. (2012). Expert voices. Expert voices. In S.E. Baker & R. Edwards. How many qualitative interviews is enough. National Centre for Research Methods Review Discussion Paper, pp. 12–14.

[3]. Baker, S. E., & Edwards, R. (2012). *How many qualitative interviews is enough? Expert voices and early career reflections on sampling and cases in qualitative research*. Southampton, United Kingdom: National Centre for Research Methods.

[4]. Bonde, D. (2013). Qualitative Interviews: When Enough is Enough. *Qualitative Market Research*, pp. 1–10.

[5]. Cenoz, J. (2009). *Towards multilingual education: Basque educational research from an international perspective* (Vol. 72). Multilingual Matters.

[6]. Hou, C.-K. (2012). Examining the effect of user satisfaction on system usage and individual performance with business intelligence systems: An empirical study of Taiwan’s electronics industry. *International Journal of Information Management*, *32*(6), pp. 560–573.

[7]. Corr, L., & Stagnitto, J. (2011). *Agile data warehouse design: Collaborative dimensional modeling, from whiteboard to star schema*. DecisionOne Consulting.

[8]. Dastgir, M., & Mortezaie, A. S. (2012). Factors Affecting the End-User Computing. *Business Intelligence Journal*, 5(2), pp. 292–298.

[9]. Davison, J., & Deeks, D. (2007). Measuring the potential success of information system implementation. *Measuring Business Excellence*, *11*(4), pp. 75–81.

[10]. Dedić, N., & Stanier, C. (2016a). An Evaluation of the Challenges of Multilingualism in Data Warehouse Development. In *Proceedings of the 18th International Conference on Enterprise Information Systems* (Vol. 1, pp. 196–206). Rome, Italy.

[11]. Dedić N., & Stanier, C. (2016b). Measuring the Success of Changes to Existing Business Intelligence Solutions to Improve Business Intelligence Reporting. In Research and Practical Issues of Enterprise Information Systems, CONFENIS 2016. *Lecture Notes in Business Information Processing* (Vol. 268, pp. 225–236).

[12]. DeLone, W. H., & McLean, E. R. (1992). Information systems success: The quest for the dependent variable. *Information Systems Research*, *3*(1), pp. 60–95.

[13]. DeLone, W. H., & McLean, E. R., 2003. The DeLone and McLean Model of Information Systems Success. *Journal of Management Information Systems*, *19*(4), pp. 9–30.

[14]. Dempsey, G. (1999). "A hands-on guide for multilingual websites," *World Trade*, *12*(9), pp. 68–70.

[15]. Denzin, N. (2012). Expert voices. In S.E. Baker & R. Edwards. *How many qualitative interviews is enough*. National Centre for Research Methods Review Discussion Paper., pp. 23-24.

[16]. Guest, G. (2006). How Many Interviews Are Enough?: An Experiment with Data Saturation and Variability. *Field Methods*, *18*(1), pp. 59–82.

[17]. Hossain, S., Islam, F., Karim, R., & Siddique, K. N. (2014). A Critical Comparison Between Distributed Database Approach and Data Warehousing Approach. *International Journal of Scientific & Engineering Research*, *5*(1), pp 196 - 201.

[18]. Imhoff, C., Galemmo, N., & Geiger, J. G. (2003). *Mastering Data Warehouse Design: Relational and Dimensional Techniques*. *Wiley Publishing, Inc.* Indianapolis: Wiley Publishing, Inc.

[19]. Inmon, B.W. (1995). *Building the Data Warehouse*, New York: Wiley.

[20]. Kimball, R. (2001). Design Tip #24: Designing Dimensional In A Multinational Data Warehouse. Retrieved from: http://kbase.com/pdf/dwdesigntips.pdf , (accesed 05 June 2017).

[21]. Kimball, R., & Ross, M. (2011). *The data warehouse toolkit: the complete guide to dimensional modelling* (2nd ed.). John Wiley & Sons.

[22]. Kimball, R., Ross, M., Thorthwaite, W., Becker, B., & Mundy, J. (2008). *The Data Warehouse Lifecycle Toolkit*. Wiley India Pvt. Limited.

[23]. Kumhyr, D. (2001). Method and system for generating and serving multilingual web pages. *Patent*. United States.

[24]. Kurbel, K. E. (2013). *Enterprise Resource Planning and Supply Chain Management*. Springer Science & Business Media.

[25]. Lepouras, G., & Vassilakis, C. (2000). Multilingual Web Site Construction and Maintenance. In *Proceedings of the SCI 2000 conference* (pp. 56–61). Orlando, Florida.

[26]. Linstedt, D., Graziano, K., & Hultgren, H. (2010). *The New Business Supermodel, The Business of Data Vault modeling*. *Lulu. com* (2nd ed.).

[27]. Miller, D. (2012). Expert voices. In S.E. Baker & R. Edwards. How many qualitative interviews is enough. National Centre for Research Methods Review Discussion Paper., pp 31).

[28]. Nebot, V., & Berlanga, R. (2016). Statistically-driven generation of multidimensional analytical schemas from linked data. *Knowledge-Based Systems*, *110*, 15-29.

[29]. Olaru, C. (2014). Business Intelligence in Telecommunications Industry. *International Journal of Economic Practices and Theories*, *4*(1), pp. 89–100.

[30]. Petter, S., DeLone, W., & McLean, E. (2013). Information Systems Success: The Quest for the Independent Variables. *Journal of Management Information Systems*, *29*(4), pp. 7–61.

[31]. Rahman, N., (2013). Measuring Performance for Data Warehouses-A Balanced Scorecard Approach. *International Journal of Computer and Information Technology*, *4*(1), pp. 1-6.

[32]. Raniner, R. K., & Cegielski, C. G., (2010). Introduction to Information Systems: Enabling and Transforming Business (3rd ed.). John Wiley & Sons.

[33]. Romney, A. K., Weller, S. C., & Batchelder, W. H. (1986). Culture as consensus: A theory of culture and informant accuracy. *American anthropologist*, *88*(2), 313-338.

[34]. Sen, A., & Sinha, A. P. (2005). A comparison of data warehousing methodologies. *Communications of the ACM*, *48*(3), 79-84.

[35]. Talburt, J. R., & Zhou, Y. (2015). *Entity information life cycle for big data: Master data management and information integration*. Morgan Kaufmann.

[36]. Vázquez, S. R. (2013). Localizing accessibility of text alternatives for visual content in multilingual websites. *ACM SIGACCESS Accessibility and Computing*, (105), 34-37.

[37]. Wang, R. Y., & Strong, D. M. (1996). Beyond accuracy: What data quality means to data consumers. *Journal of management information systems*, *12*(4), 5-33.

[38]. Warren, J., Benbow, H. M. (2008). *Multilingual Europe : Reflections on Language and Identity*. *Cambridge Scholars Publishing*. Newcastle upon Tyne, UK: Cambridge Scholars Publishing.

[39]. Dedić, N., & Stanier, C. (2017). Measuring the success of changes to Business Intelligence solutions to improve Business Intelligence reporting. *Journal of Management Analytics*, *4*(2), 130-144.