Serving Massive Time-Based Media (SMaTBaM)

Report on the evaluation process of system needs and demands of serving time-based media in the area of the Performing Arts

Document Information

Title:	Serving Massive Time-Based Media
Subtitle:	Report on the evaluation process of system needs and demands of serving time-based media in the area of the Performing Arts
Authors:	Stephen Malloch, Carola Boehm
Authors'emails:	carola@music.gla.ac.uk, mallocsc@music.gla.ac.uk
Date:	June 97
Contact person:	Dr. Stephen Arnold
Email:	stephen@music.gla.ac.uk
URL:	http://www.music.gla.ac.uk/HTMLFolder/Research/smatbam-private/index.html

Content

1. Description and Objectives

Brief Description

The SMaTBaM[1] project (Serving Massive Time-Based Media) aimed to establish a working example of a dataserver for time-based media in the area of the Performing Arts. Building on experience gained in the UMI/NetMuse project[2], the project will demonstrate the searching and serving of massive realtime data, using a web-based interactive front end. The project is designed directly to enhance the Performing Arts Data Service, PADS[3] , whose remit and funding from the Arts and Humanities Data Service, AHDS[4], do not extend to conducting research into the development of specialized supporting technology. (PADS is a Glasgow University[5] Arts Planning Unit[6] Project based in the

[1] The SMaTBaM project is funded by the Scottish Higher Educational Funding Council, http://www.shefc.ac.uk/shefc/welcome.htm

[2] http://www.music.gla.ac.uk/HTMLFolder/Research/NetMuse.html

[3] http://www.arts.gla.ac.uk/PADS/

[4] http://www.kcl.ac.uk/projects/ahds/top.html

[5] http://www.gla.ac.uk

[6] http://www.arts.gla.ac.uk

Departments of Music[7] and of Theatre, Film and Television Studies[8].)

Objectives
1 - to develop a server system for time-based media, with direct access to the MAN
2 - to develop a "pilot" dataset of time-based materials
3 - to develop a generalized search and retrieval interactive browser, based on the www model
4 - to consider representational models for time-based media (report)
5 - to consider analytical tools for time-base media (report)
6 - to develop strategies for securing research funding in this area on a long term basis (report)
7 - to present findings to the community via the AHDS[9] and other relevant fora and publications

2. Providing a Service

Today, research centres, libraries, universities and the public in general wish to access and use the best information and data possible. This trend is emphasised by a tremendous need for user friendly and flexible systems with advanced information retrieval capacities for research an teaching.

Time-based media, as for instance the performing arts with music, film, video, theatre and dance, have their specific demands which have been dealt in past projects only to a certain extent. The inherent character of their time-based content and the aim to facilitate real-time access in the highest quality possible provides problems that only high performance servers and networks can cope with, their inherent character of their multimedia content creates archiving problems that cannot be dealt with the traditional relational database model or catalogue systems.

Though much of archival material to be stored and distributed will inevitably always be simple data types, such as text, image, audio or video rather than more complex or composite types of data, appropriate mechanisms for searching and standards for exchanging information efficiently are needed. Furthermore structures and models are needed to fulfill the needs of the more complex relationships between these types of data.

The use of digital data resources to facilitate research and teaching in the performing arts has to define methods of storing and distributing complex time-based data to be able to serve quality AND quantity information across wide area networks.

3. Requirements and Definitions

3.1. Description of the Resources

It is envisaged that a collection dealing with time-based media will consist of both secondary resources, i.e. materials about the performing arts, moving image and sound-based media, and primary resources, i.e. the digitised multi-media objects themselves. As data compression and transmission technologies develop in the future, it should be a service's aim to facilitate the real-time access of video clips, sound files, movies, musical performances and multimedia productions.

The collection should be able to be expanded by collections of other service providers holding resources in the same field and maintaining a one-stop shop in accessing time-based media resources. This distributed resource environment, in the best case, will let other collection holders keep and maintain their collection in their own repository, while access will be handled by a central access point.[10]

A time-based resource collection encompasses a wide range of different disciplines, starting with the disciplines of

---

[7] http://www.music.gla.ac.uk

[8] http://www.arts.gla.ac.uk/

[9] http://www.kcl.ac.uk/projects/ahds/top.html

[10] The National Preservation Office of the National Library of Australia has made this "Distributed responsibility" as one of its Statement of Principles of Preservation of and Long-Term Access to Australian Digital Objects. See http://www.nla.gov.au/3/npo/natco/princ.html

music, art and video and stretching further toward dance, theatre and the broadcasting arts. A first taxonomy[11] of the scope of the prospective resources displays the variety and discusses different means of structuring the resources into sub groups.

3.1.1. Different types of data

As can be seen from the taxonomy, all the "usual" multimedia data types are evolved from sound, video, text, image and binaries. Storing them in a certain way provides us with a more complex entity of data types: html, sgml, mpeg, wav, gif, jpeg, java, etc. It is certain that these data types will evolve further in number and content. The use of different data types in a system should therefore be a means but not a solution. In other words, to minimise the danger of storing data in standards that might not be supported in the future, much thought should be taken into separating the content of a resource from its presentations. To be able to store a resource in its highest quality possible, combined with the ability to convert it into representations, efficient for a certain usage, or added representation in the future, is to provide an open and flexible system with maximum compatibility in long terms.[12]

3.1.2. Different complexities

Whereas video and art might be stored largely as single binary data-objects, music, theatre and the broadcasting arts could envolve the storing and accessing of highly structured data, presenting complex objects or 'composite objects'.[13] In some cases, it might be hard to distinguish which is the real, the original resource, and which is a composite part of it. If one accepts the fact that the content of a resource might be of complex or composite nature, then the step towards devising a way to store it as such is not far. Technologies are needed that offer the ability to depict, represent, access, store and manipulate complex structures in their complex "Gestalt". A broadcasting feature, as one resource, might encompass video data, sound data, and text data and still be one work of art.

We should accept the fact, that our future data might not remain in its binary form and much of our present resources have never been in the "Gestalt" of one entity. Java Applets, Webobjects and other distributed object environments are already being used by artists to create an opus made out of many components and having many facades to it. Also the existing resources, which have been traditionally stored as metadata in catalogues, while their real content is being stored as artifacts in shelves, cassettes, or discs, are often not just one entity. In trying to devise resource systems of the next decade, it would be illogical to diminish the resources and their "real-life" manifestation by disregarding their composite character.

3.1.3. Different Relationships

"Knowledge can be defined as pieces of information put in context to each other".

Assuming that we have objects stored in a persistent way, the access and search results are influenced by the context these objects are in. The mapping of content and context into a digital world means to define and store different kind of

---

[11] See Categories of time-based Media, http://www.music.gla.ac.uk/HTMLFolder/Research/smatbam-private/categories.html

[12] 3For an example of the separation of content resource and various representational views, one might think of a picture stored in the highest resolution possible in a central resource archive, and its conversion to gif, compressed, lower quality for web use. When using high-speed networks, one would still be able to use a higher quality resource to provide users with, or an even lower one due to any possible copyright restrictions. In the computing world, this separation of content and representation has one of its object-oriented manifestations in the Model-View-Controller paradigm. The model being the content, the data, or a knowledge domain, the view being one possible presentation of it. The controller can be seen as the gadget maintaining the connection between the model and the view. One note, for instance, could be depicted in a system by an internal, proprietary data structure. To this note, one or more views can be "plugged in" as for instance a midi representation, a sound representation, a graphic representation. Devising new views is thus independent of the content. See also Jacco van Ossenbruggen: Music in Time-Based Hypermedia, http://www.cs.vu.nl/~jrvosse/Papers/echt94/htmlindex.hltm

[13] Elementary or simple objects = objects made out of one entity or one binary (text files, bitmaps, wave format files, midi files) Composite objects - composite objects consist of a number of elementary or composite objects, for instance a complex/composite music data structure Complex objects - objects with attributes, that change in size

relationships between objects[14] Relationships can be of numerous variety. Some examples:

Five relationships already widely used in information systems are

- **Inclusion** - one object is included in another object (ex.: a file in a folder, a certain sound used in a composition a note in a bar)
- **Inheritance** - one object inherits the characteristics of another object (ex.: all Bach works have a BWV, so each single works inherits the attribute BWV-verzeichnis-number of the Bach Works Object; or, all service provider users have read rights, these might be inherited down towards the developers of collections, who also have write rights; or, as a third example, all sounds stored in a high quality inherit the characteristic of being served out only on ATM in real-time)
- **Association** - one object is associated with another object (ex.: Mendelssohn "Fingals Cave" , the composition resource, is associated with the resource of geographical rock formation of Staffa. Another example would be that two pages can be associated with each other in form of sequence. One page should follow the other in a certain context as for instance a book, course, slide show, etc)
- **Attributes** - an object contains certain attributes, or certain characteristics which describe its state of being or its internal structure (ex.: all objects in the PADS archive have the attribute DublinCore, where the DublinCore object itself has 15 further attributes defining the elements of the Dublin Core
- **Web Links** - Web-links can be thought of being a realisation of a certain kind of association

The publication of these resources involves the presenting of one resource via different types of other resources or one resource related to others. For instance, a computer-music piece may be present in its presentation as a sound file, presenting the first recorded performance, as well as archived as the code of the computer program itself and the secondary information associated with this resource.

3.1.4. Time-based media

The common denominator of all the prospective resources of the SMaTBaM and PADS projects is the characteristic of being time-based. Just as the human mind has to receive and combine and remember information differently, depending on whether the artefact looked at is "just" two-dimensional or if it contains an additional third dimension, thus having a "fleeting" characteristic, storing and accessing time-based media requires special attention in storage and delivery of the objects.

Solutions are needed to store information in its inherent complex form on the server side, to transmit these information packages in real-time with high-quality over a wide-area network, and to provide a user interface to be able to access and use the resources intelligently.

For a high-quality service four types of time-based material, all requiring real-time access, can be identified:

- **Large binary data objects**: as sound or video - streaming binary data combined with using a guaranteed bandwidth to ensure no glitches or breaks. Requires:
  - ATM networks (or any networks providing high bandwidth and guaranteed quality of service)
  - client-server software tools to provide the streaming
  - high-performance media servers
- **Subsets of large binary data objects**: playing just a part of a sound or video
- **Two or more parallel large binary objects**: such as synchronisation of multiple audio streams, requires
  - Intra-stream and Inter-stream synchronisation to maintain the temporal relationship between multiple streams. As for instance in the case of 'lip sync' in film and tv, where sound and vision tracks are often

[14] See Relationship Service Specification for distributed objects, (OMG) http://www.omg.org/corbserv/relation.pdf and also the work done by the Laboratory for Advanced Information Technology: the Knowledge Interchange Format, KIF, http://www.cs.umbc.edu/kse/kif/kif101.shtml

recorded on different media.[15]

- **Complex objects**: such as MAX music scores, more complex Java applications, or sound-sound combinations, require a fast and time-coordinated access of all the composite parts of an object: the synchronisation of multiple, periodic, logically independent streams of arbitrary type. [16]

3.2. Definition of the users

In a digital resource collection service there are different types of users. This has to be taken into account when implementing secure access with rights management, i.e. the maintaining and administering of access rights and licenses.

From experiences gained in the NetMuse and the MusicWeb projects, time-based media collection owners, such as museums, music publishers and labels, will only be willing do offer the use and the publishing of certain collections if restricted access can be guaranteed, and sometimes only if the management of the rights is transparent and administered by the collection owners themselves. The dependency of these bodies on their collection as a means of income, as well as the maintenance of existing licenses and access rights of their collection, might also be a important issue when discussing collection input.

User levels and rights management that can be identified are:

USERS	RIGHTS MANAGEMENT
Providers • Developers • Collection Officers • Project Managers	**Providers** • full access, configuration and programming rights • full access – • full access
Close associates • Content owners and providers	**Close associates** • full access and management rights to parts of a collection
End Users • user groups with rights to certain collections • anonymous users	**End Users** • read access to part of a collection • read access to public collections

3.3. Definition of the delivery

In the ideal scenario delivery bandwidth and quality of service would be more than adequate for the delivery of as many high quality audio or video streams are desired. However, although the bandwidth trends are (like processor speeds and storage capacity) towards higher and higher capacity, excessively large bandwidth data such as uncompressed ITU-601 video means that existing high bandwidth networks could quickly be swamped with a very few number of streams. Compression therefore will play a part in any media serving system to be developed within a realistic time frame.

For video, the issue of quality of picture will be discussed elsewhere but cannot be disassociated from the definition of the delivery as quality has a direct effect on delivery bandwidth. For high-quality, both pictures and sound, some kind of high performance networking is required even when using compression. Audio has stricter requirements for quality of service, but less on bandwidth; whereas video has a greater requirement on bandwidth but is more tolerant of the occasional dropped frame. (The eye is more tolerant than the ear...)

ATM networks such as the UK MAN's (Metropolitan Area Networks) provide a suitable platform for delivery of high-quality media streams. It is not unrealistic to assume that these MAN's could one-day be connected via high-speed interMAN links providing a UK wide environment where high-quality streaming is possible. On the other hand, however, it is not reasonable to assume that everyone who could benefit from the teaching and research resources available in a

---

[15] See Geroge Robertson, MIniMS, Multi-Participant Interactive Music Services, http://www.dcs.gla.ac.uk/~george/minims/minims.html and George Robertson, Sample Rate Synchronization across ATM Network, Proceedings of the ICMC, Thessaloniki 1997.

[16] See Scott Flinn: Coordinating Heterogeneous Time-Based Media Between Independent Applications, http://www.cs.ubc.ca/spider/flinn/publications/mm95/scheduler.html

service (such as PADS) will have direct access to these MAN's, or indeed, any high-performance networking.

In the best realistic case then, a "scalable service of delivery" should be enforced, meaning the serving of data in a quality and quantity corresponding to the performance of the network connection. Although the development and research in serving time-based media is to be concentrated on the highest quality, i.e. real-time serving over ATM networks, the resource collections should additionally be accessible through lower performance networks.

3.3.1. Connection

High-quality service will be realised with ATM networks, necessary to maintain a guaranteed quality of service, required for handling time-based media. For connection on ATM networks, specialised client software, realised as stand-alone helpers or plugins to mainstream browsers, will ensure the real-time streaming of large-sized audio and video streams. Implementation paradigms, such as splitting content from representation, and an object-oriented or modular system architecture will enable the delivery of resources in different levels of quality. Thus without having an ATM connection, the user will be able to access the same sound with standard audio and video players, but without being able to play it in real-time. The same procedure will be valid for more complex data, for which the aspect of time is of relevance.

3.3.2. Access tools

Having a resource collection in place, there might be various means of accessing the data. On the most general level, the use of a normal web-browser for accessing objects in the resource collection should be enabled. Prerequisite is an interface handling the automatic web page creation out of the objects in the database. This web-gateway should have an open system architecture, enabling to add and change web page elements in various formats, such as sgml, html, xml. In the best case, html elements, or xml elements, should be objects or modules themselves in the system, thus being able to expand the generation of web pages by future web authoring standards.

As mentioned above, on a second level, there will be proprietary tools to access the objects in the highest quality, as for instance real-time sound might be streamed over ATM networks with a proprietary audio-player, which has a client/server architecture itself.[17]

Beside the access tools of the enduser, there should also be a database client, with full programming and access rights to the system itself.

- Proprietary plugin viewers for highest quality (streaming audio/video ex. NetMuse audioplayer)
- Database client (for programming, administering, and accessing full database functionality)
- Web-browser

QualityConnection throughUser access through	
high • (guaranteed quality of service)ATM (native) • proprietary client software for media players • usual media plugin client software supporting ATM protocolsmediumTCP/IP over ATM - near to best quality (realistic today) • usual media plugin client software • (optional database client either proprietary or realised as a browser)	**low** • TCP/IP • ISDN • analogue modemusual media plugin client software

3.4. Database Interoperability between collection-holders

[17] A Java audio-player in client-server architecture , streaming audio over ATM has been developed in the NetMuse project at the University of Glasgow, Department of Music. http://NetMuse.gla.ac.uk/

"Our feeling is that at this point "metadata" as a descriptive term has become so debased by overuse (and means so many different things in different communities and contexts) that it is now virtually meaningless without extensive qualification; unfortunately, it has also become a very fashionable term"

Lynch, Michelson, Summerhill, Preston "The nature of the NDIR Challenge"

http://www.cni.org/projects/nidr/www/chapter-1.html, June 97

A goal of the PADS service, to which SMaTBaM's research if orientated, is to provide interoperability with other collection holders by conforming to and implementing relevant standards. To shortly sketch the status-quo situation of using multi-media digital resource collections already available, one can look towards broadcasting stations, music/video archives, record companies and libraries. It must be taken into account that collections are stored in different storage mediums, ranging from simple file systems, to relational database management systems to the growing number of object-oriented database management systems.[18] In addition to that, a large number of music-catalogues in a variety of formats has to be also made accessible.

Between catalogues, an implementation of the Z39.50 protocol, version 3 from 1995 [19] will be sufficient. For interfacing catalogues with relational databases, there will have to be a Z39.50 - SQL interface. There have been very few relational database vendors who have implemented a Z39.50 support. Only one reason being that their "interoperability protocol" has been SQL, which has been universally accepted and implemented by almost all of the database vendors.

There have been already discussions in the past to extend the Z39.50-1995 protocol with SQL.[20] From here it is just another step and a matter of time, to stay interoperable with the present database generation, which are based on object-oriented technologies and have defined an object query language (OQL.) and object definition language (ODL).[21] With the prospective wide-spread use of digital libraries, OODBMS, or object-oriented database management systems will become to be a major means of storing, accessing and using complex, multimedia data objects.

I refer to research works and projects, which have been influenced largely by projects in cooperation with the Library of Congress, amongst them Kahn/Wilensky Framework for his Digital Object Architecture , A Framework for Distributed Digital Object Services,[22] Daniel Lagoze's Dienst/NCSTRL[23]-and The Warwick Framework A Container Architecture for Aggregating Sets of Metadata.[24]

Assuming a main interoperability of different collections holding digital, multimedia objects, the underlying transfer protocol will have an influence on the performance, quality and representation means of the objects to be delivered. Using a stateless protocol, such as http, means that only one object can be delivered per session. Thus the connection closes after each document is delivered, losing all the information of the former session.

---

[18] See examples: Time-Warner Pathfinder Personal Edition, http://pathfinder.com/@@5cnHOgcAhVYFFeXJ/welcome/ a personal magazine, The Chicago Tribune's Metromix , http://www.metromix.com, EDS, Aniamtion 200 , Liberation (Libraries: Electronic Remote Access to Information Over Networks, http://www.iicm.edu/liberation, Frequency Banana

[19] Library of Congress Maintenance Agency page for International Standard Z39.50, http://lcweb.loc.gov/z3950/agency/

[20] Proposal for SQL Access in Z39.50: Z39.50/SQL+, http://www.dstc.edu.au/DDU/research_news/reports/zproposal.html

[21] ODMG 2, http://www.odmg.org/

[22] Kahn/Wilensky, Digital Object Architecture. A Framework for Distributed Digital Object Services, http://www.cnri.reston.va.us/home/cstr/arch/k-w.html

[23] Daniel Lagoze, Dienst/NCSTRL, http://www.ncstrl.org/

[24] Daniel Lagoze, The Warwick Framework A Container Architecture for Aggregating Sets of Metadata, http://cs-tr.cs.cornell.edu:80/Dienst/Repository/2.0/Body/ncstrl.cornell%2fTR96-1593/html

In devising a secure and distributed system, with collections stored in different locations, access handled from a central gateway and user access in the best case being controlled to a point of write, read and execute rights of single objects and collections, stateless protocols can be a problem. Solutions lie in the underlying existence of user rights management, such as a database management system able to control the access of many users in dependency of objects or collection of objects, or/and the use of a stateful protocol such as Z39.50 or hyperg.

All in all, the key goals of a provider serving time-based media will be providing interoperability through

- a Z39.50 gateway,
  - implemented as a gateway between the database management system and the user/client interface through either SQL or OQL or a query language appropriate for the database management system
  - implemented as an additional access path to indexed files, which are created dynamically from the
  - attributes (Metadata) of the objects stored in the database management system
- a SQL compliant interface
  - to be able to access other collections held in relational database management systems
  - to be able to be accessed through the Z39.50 gateway (see above)
- a OQL or a query language appropriate for the database management system
  - to be able to access objects in terms of their characteristics (attributes and functionality)
  - (in case of ODMG OQL) to be able to access objects from other object-oriented database management systems

3.5. Digital Resource Preservation for time-based Media[25]

As the paper published by the JISC/BRITISH LIBRARY Workshop of the 27th and 28th November 1995 at the University of Warwick states, there are three types of digital resource preservation: medium preservation, technology preservation and intellectual preservation.

"The problem, and what is new about preservation in the electronic environment, is that electronic information must now be dealt with separately from its medium. This can be illustrated by an analogy, one which is very oversimplified, as readers will be aware: if a book is placed on a closet shelf, and the closet door is closed for 500 years, then at the end of that time one can, broadly speaking, open that door and read the book. With an electronic resource one does not have that confidence after ten years, and for several reasons."[26]

In the case of having digital resources as the resource itself (not only having records or catalogues describing it) we have, as touched above, the content and the representation of a resource. The content is the resource itself, the information of a knowledge domain. The viewers is the means for the user to see or access this information, thus the applications, or the representation of the knowledge domain. Example would be a piece of digitally stored music "viewed" with a Netscape sound player, or a Real-Audio streaming player, or a CD player. The content of the music seemingly does not change. The viewers, or in other words the representations, do change.

In devising systems in which the rapid changes of technologies will not make the means of viewing information obsolete, it is required to implement a separation of content and view as much as possible. The traditional technique of archivists was the "refreshing" of digital information by copying it into a new standard, a new media, or a new format or "migrating" it from one hardware/software configuration to another.[27] Both techniques can be lossy and time consuming. In adapting systems with the separation of content and viewer, combined with the ability of plugging in new viewers, a maximum of independence of technology change is achieved, while the resource is digitally stored in the highest quality possible and remains as that. If compression methods are needed to solve any storage shortages, then a

---

[25] See Sites about Digital Research Preservation: AHDS, Digital Research Preservation in the AHDS, http://www.kcl.ac.uk/projects/ahds/preserve.htm, Bibliography of Resources Related to Preservation of Digital Resources http://www.oclc.org:5046/~weibel/archtf/archtf_bib.html LONG TERM PRESERVATION OF ELECTRONIC MATERIALS A JISC/BRITISH LIBRARY http://ukoln.bath.ac.uk/fresko/warwick/intro.html Preserving Digital Information: Final Report and Recommendations http://www.rlg.org/ArchTF/

[26] LONG TERM PRESERVATION OF ELECTRONIC MATERIALS, http://ukoln.bath.ac.uk/fresko/warwick/intro.html

[27] The Commission on Preservation and Access and The Research Libraries Group, Inc., The Challenge of Archiving Digital Information, http://www.rlg.org/ArchTF/tfadi.chelng.htm

losslessly compression method, or a compression method with the least loss, should be used.

In the present era of digital distributed resources, most of the time a storing format is used, that will enable the most efficient delivery. This results in using compression methods, that would be unacceptable for academic research or cultural heritage preservation. Delivery means and storage means will have to be separated in digital archives and libraries, moving into the era of using digital resource preservation as a means to archive cultural heritage. Even if this seems an unrealistic viewpoint from todays' situation, we will have to deal in a few centuries with digital artifacts that cannot be reconstructed anymore into their original quality, due to the storing in a compressed, lossy formats. Taking museums and archives for a model, their main aim is to preserve the artifacts as best as possible, and most of the financial expenditures of museum is allocated to this preservation of a cultural heritage.

Summing up: aims, that should be taken in a system design to guarantee a high quality of digital resource preservation:

Separate content from representation

Use different storage formats than delivery formats

Use losslessly compression methods for storing if possible

The third preservation requirement, specified in the paper published by the JISC/BRITISH LIBRARY is intellectual preservation, addresses the integrity and authenticity of the information as originally recorded. To sum up the changes a digital resource may undergo I follow the JISC/BRITISH LIBRARY Workshop:

Accidental change (data loss during transfer, accidents during updating, saving the wrong version)

Intended change / well meaning:

New versions or drafts (authorial texts, legislative bills);

Structural changes (updating Books In Print or a telephone directory);

Interactive documents, (hypertexts with note-taking capabilities)

Intended change / fraud: (political papers, laboratory notebooks, historical rewriting, legal documents, contracts)

This preservation aspect rather addresses security, versioning and copyright issues and has to be handled by a system as such:

The digital resource in its resource archive must be secure of unwanted changes, i.e. a secure rights

administration on the level of user groups and object collections

The digital resource may change over time, i.e. a versioning scheme might be required.

The digital resource must have a way of identifying its copyright-holders in a secure way, i.e. either the copyright

owner information has to be imprinted on the resource itself, or must be attached to it in a secure manner

4. User Requirements for Time-Based Media

4.1. Platform and Performance Requirements

The major issue for a user accessing a resource of time-based material is to get quality. Although this seem to a certain degree to be subjective, a successful service should at least be able to serve material in the quality a user is accustomed to. Additional to that, a lower quality service should be possible for low-end users. Preferably, a scalable service of delivery should enable the user to weigh quality against efficiency of his own needs and hardware specifications by himself.

High-quality delivery (Assuming the availability of at least 25Mbits/s ATM networking to the desktop)

Audio: Instant CD-quality service, 16bit, 44.1kHz digital audio

Video: Instant digital TV/Video quality, MPEG2 (compressed broadcast quality CCIR-601 images at up to 15

Mbits/sec)[28]

Low-quality delivery (ethernet connection)

Audio: (downloadable) 8bit 16kHz

Video: (downloadable) MPEG 1 (compressed SIF images approximate to VHS quality at 1.5 Mbits/sec)[29]

An example of a hardware/software specification / PC for High-quality delivery:

Windows 95 or NT.

Processor: Pentium/Pentium Pro. At least 133Mhz; 180Mhz or 200Mhz

Hard disk - over 1Gb, SCSI drive/adapter better than IDE/EIDE drive, At least 32MB RAM

17 inch monitor (1280x1024 resolution)

Graphics card supporting 64,000 colours at the 1280x1024 resolution (as for instance the Matrox Millenium card)

Sound card (digital output recommended for high quality audio.)

External/powered speakers (to maintain CD quality audio, quality external amplifiers and speakers (and DAC) are recommended.)

ATM adapter card, (for desktop ATM connection to min. 25Mbits/s ATM network) which supports native ATM as

well as IP over ATM (as for instance Fore Systems ForeRunner network adapter card)

(Ethernet card - for access to local networks)

Client-side software plugins (media-players)

4.2. Providing Time-Based Media in Realtime

For the levels of real-time access identified in chapter 3, there might be different end-user platform and performance requirements.

- **large binary data objects**
  - streaming media players on the client side, connecting to a media server on the server side. Realized
  - either as webbrowser plugins or as database-client software modules (if database client is available on a
  - wide-area-network) example: the Netmuse audio player.[30]
- **subsets of large binary data objects**
  - Media players should support input parameters as starting point and end point of an audio/video file
  - example: Netmuse audio player[31], HyperwaveAudioViewer, etc.

---

[28] Deriving from the CCIR-601 digital television standard which is used by professional digital video equipment. It is (in the US) 720 pixels/line by 243 lines by 60 fields (not frames) per second, where the fields are interlaced when displayed. The chrominance channels are 360 by 243 by 60 fields a second, again interlaced. This degree of chrominance decimation (2:1 in the horizontal direction) is called 4:2:2. Although other video formats can be used as input to MPEG-II compresion some form of CCIR-601 (also known as ITU-601 or D1) video is normally used - hence the improvement over MPEG-I. (see http://www.cis.ohio-state.edu/hypertext/faq/usenet/mpeg-faq/part0/faq.html)

[29] The source input format for MPEG I, called SIF, is CCIR-601 decimated by 2:1 in the horizontal direction, 2:1 in the time direction, and an additional 2:1 in the chrominance vertical direction. Some lines are cut off to make sure things divide by 8 or 16 where needed. The SIF video is then compressed by the MPEG-I system using a compression ratio of about 26:1. MPEG-I bit rate is 1.5Mbits/sec, typically made up of 256kbit/sec of compressed audio and about 1.15Mbits/sec video. Standard also allows lower bit rates. (see http://www.cis.ohio-state.edu/hypertext/faq/usenet/mpeg-faq/part0/faq.html)

[30] Netmuse, http://www.music.gla.ac.uk/HTMLFolder/Research/NetMuse.html

[31] Netmuse, http://www.music.gla.ac.uk/HTMLFolder/Research/NetMuse.html

- **two or more parallel large binary objects**
  - Time-Manager/Synchronization tools on the client side
  - example: planned synchronization package from MiniMS, Multi-Participant Interactive Music Services,[32]
- **complex objects**
  - possibilities of maintaining time-based structures with complex and composite objects implies
  - object-based time-management, realized either in a distributed way as an ORB or in a server/client side
  - way as in OODBMS. These possibilities have yet to be researched.[33]
  -

5. Service Provider Requirements

5.1. Platform and Performance Requirements

- the system must run with acceptable performance on a server with a hardware specification sufficient to meet the access requirements listed below, allowing room for expansion. (It is unlikely such a workstation would be less specifed than a Sun Sparcstation 10/512, 2 x 50 MHz proc., 128 MB RAM or a Silicon Graphics Challenge Server R5000, 180 MHz, 128 MB and may be much more highly specified. It also unlikely that the server will be running a non-unix variant OS.)
- the system should have a client-server architecture
- system arcitecture should in the best case be distributable and platform independent
- system architecture should be scalable to cope with significant increases in content (volume of material), registered and anonymous users, number of servers in the system and number of objects in the system.
- database client and database server should be able to run on one machine, as well as on different machines of different platforms

Numbers estimated for up to the next 3 years and valid for one server system (in a possible server pool)	
Number of simultaneous user sessions (database client / collection manipulation)	30
Number of simultanous hits on the web gateway	5000/day
Number of registered users	500
Number of minimum objects in the database	1 * 10e6
Number of minimum documents of average size (web documents)	200 000
Number of simultaneous streams of one object	10
Number of simultaneous streams of different objects	limited only by network

5.2. Interoperability

- Metadata/DublinCore compliant object attributes

---

[32] http://www.dcs.gla.ac.uk/~george/minims/minims.html other research work in this area: Scott Flinn , Coordinating Heterogeneous Time -Based Media Between Independent Applications, http://www.cs.ubc.ca/spider/flinn/publications/mm95/scheduler.html

[33] Jacco van Ossenbruggen, Anton Eliëns: Music in Time-Based Hypermedia, http://www.cs.vu.nl/~jrvosse/Papers/echt94/html/ and Brian Nielsen, Paradigms and Environments for the Development of Distributed Realtime Systems, 1994. http://www.iesd.auc.dk/~bnielsen/Phd/phdproject.html

- a Z39.50 gateway
  - implemented as a gateway between the database management system and the user/client interface
  - through either SQL or OQL or a query language appropriate for the database management system
  - or implemented as an additional access path to indexed files, which are created dynamicly from the
  - attributes (Metadata) of the objects stored in the database management system
- an SQL compliant interface
  - to be able to access other collections held in relational database management systems
  - to be able to be accessed through the Z39.50 gateway
- an OQL or a query language appropriate for the database management system
  - to be able to access objects (in case of ODMG OQL to be able to access and be accessed from other
  - object-oriented database management systems)

5.3. Archive administration

- the collection should be able to be administered by the collection holders, with a secure and user friendly graphical user-interface.
- remote adminstration over a wide-area-network should be possible
- the implementation and design of collections architecture and database schema should be aided with user-friendly graphical interfaces to enable collection holders to create their own database architecture depending on their collection characterisitcs
- backups of the resources, or parts of, should be automated and updated at least once/week
- the backup should be possible with the system staying on-line
- in case of having a bigger and valuable collection in terms of valuable cultural heritage preservation, an automated backup or mirroring of an on-line archive system should be possible.
-

5.4. Scalability

System should be able to be expanded by

- additional servers
- additional users and user groups
- additional collections
- additional storing space
- additional storage mediums
-

5.5. Supportability and Compliances

To secure issues of interoperability and supportability into the future, the following standards and agreements should be supported:

- Z39.50-1995
- Warwick Framework*
- Dublin Core
- OMG *
- ORB
- ODMG93 *
- OQL, ODL
- SQL92

(*) these compliances are not necessarily needed to run a service efficiently at the present, but will be important, if not necessary, to stay compliant with future systems for digital resources

5.6. Security

Robust security features control access to server, documents, objects and structural elements based on user authentication and sophisticated access rights.

6. System Architecture Requirements

6.1. Multimedia Support

- multimedia classes
- audio and video support
- streaming support - support of high performance networks
- multiple database server architecture with multiple clients (not necessarily distributed database architecture)
- server should be multi-threaded
- scalable streaming support

6.2. Composite / Complex objects

- storing and delivery of composite objects
- storing and delivery of complex objects,i.e. objects with attributes of changing size (and better, of changing ypes)
- remote executio n

6.3. Network

- TCP/IP Network support
- ATM

6.4. Managment and maintenance

- on-line maintenance
- remote maintenance
- continuous operation whiile adding a collections, adding servers, administering system, migrating data, to
- updating system to a new version

6.5. Object and Document server

- at least a Cache Location in server, better additionally in client,
- better server and client cache expand automatically if needed
- Index database creation of objects' attributes?
- dynamic expansion of database
- Attribute inheritance
- global OID's

6.6. Wide Area Network Suppport

- Support of all Mime types
- Support of Web page generation, but with self defined web page source code language elements (HTML, SGML,
- HIF, XML, Java Script, etc)
- Support of creating and defining "template" webpage views for parts of collections
- Support of Java

6.7. Storage and Delivery

- if single server, multiple client: database volumes on multiple drives should allow parallel I/O

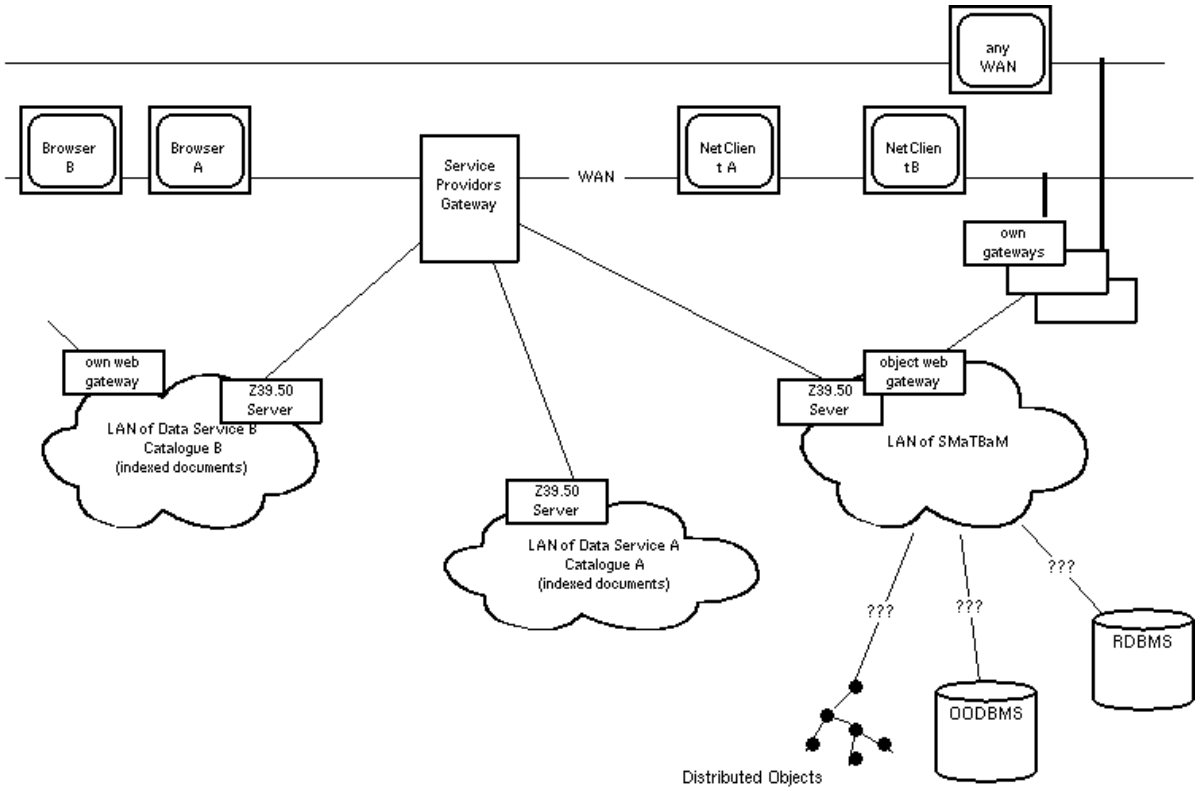
6.8. Navigation

- search/retrieve and browse collections
- graphical representaton of database content
- intelligent search and access tools

7. Proposed System Architecture

The system will be support a service provider defined in the chapters above. The service provider SMaTBaM may be one of many service providers, all compliant with Z39.50 protocols. A central Z39.50 gateway, to which the SMaTBaM system will be interoperable with, will be responsible of the managing of incoming requests. Additional direct requests

for subject specific searching will be realized in the SMaTBaM system.



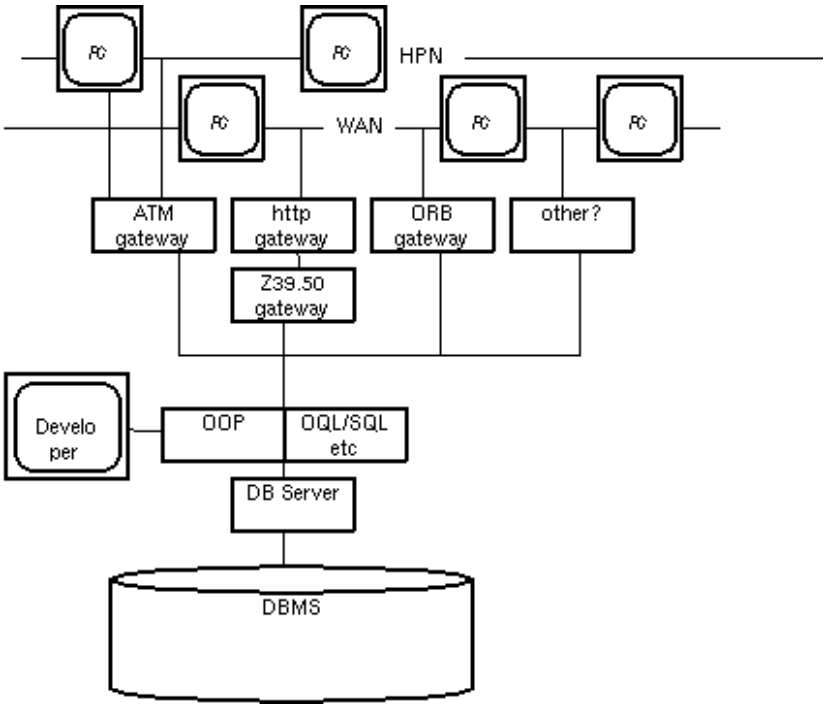
Greater context of the SMaTBaM Service

The system itself will be compliant with standard interoperability database protocols and should be able to access, relational database management systems as well as object-oriented database systems.

7.1. Basic Architecture

The basic system architecture will contain

- some kind of object-based or object-oriented database in the centre
- some kind of gateway to serve and connect the wide-area-network community
- some kind of protocol-conversion-layer or gateways support interoperability with other services
- some kind of gateway enabling the streaming of BLOBs to ethernet/ATM networks



The research into different available systems has shown, that a fast result with satisfying these demands can only be achieved by employing a multi-vendor solution.

7.2. Serving over ATM networks

Many possible system designs with different protocols for streaming over ATM networks have been investigated and researched into.[34] One possible realisation and an implementation responsible for the streaming of video and audio over ATM networks has been realised with Java by Stephen Malloch and Thorsten Pflicke at the University of Glasgow[35] On the client side, platform dependant modules are responsible for ATM networking and sound in/out functionality establishing direct connections to server-side modules. Using this client-server architecture in devising browser plugins and/or stand-alone applications for audio and video streams, the limitations of TCP are bypassed by using the IP/ATM layer. Support of the IP protocol is widespread amongst ATM hardware vendors. Vendors have developed ways of integrating IP transparently and the bypassing of the TCP protocols is possible. The direct connection to the servers, in this case of high-performance media servers, without any bottlenecks in between (as for instance a web-gateway of a database) establishes the potential full exploitation of its performance.

7.3. Serving Time-Based Media with ATM and OODBMS, one proposed database system

One architecture chosen and tried as one of the possible solutions was the following. Advantages of the system is a very fast learning curve, a optional use of a session-based protocol, and less implementation work . Major aspects of

---

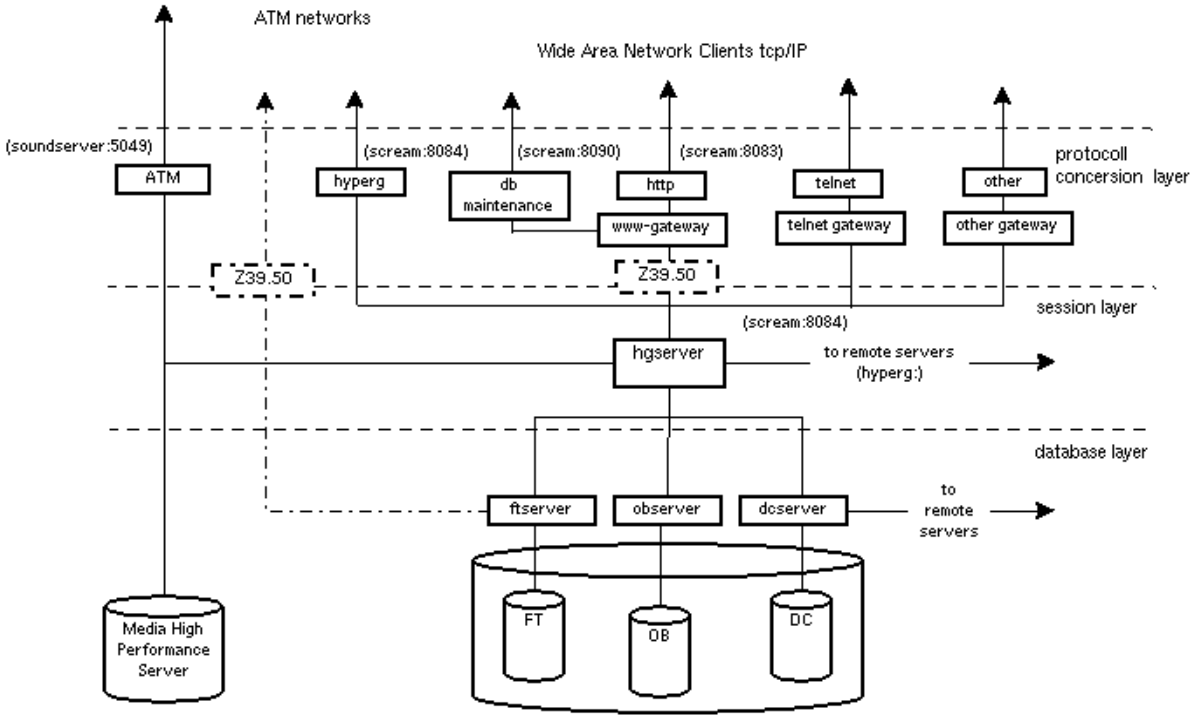
[34] Proceedings of the 20th Local Computer Networks LCN '95, at http://www.computer.org/conferen/proceed/lcn95/toc.htm and
Hermann Hueni, Beat Keller, Ein OO-Framework fuer Netzwerkprotokolle, Objektspektrum Freburar 1997, p.51-57. and
Hermann Hueni, r. Johnson, R. Engel, A framework for network protocol software, in: Proceedings of OOPSLA-95, 1995. ftp://st.cs.uiuc.edu/pub/patterns/papers/conduits+.ps and
Dourglas Schmidt, Tim Harrison, Ehab Al-Schaer, Object-Orineted Components for High-Speed Network Programming, USENIX Conference on Object-Oriented Technologies, Monterey, CA, June 1995. http://www.usenix.org/publications/library/proceedings/coots95/schmidt.html

[35] Stephen Malloch, Stephen Arnold, Torsten Pflicke, Using Java to stream audio over ATM, Proceedings of the ICMC 97.

the operational requirement are already partly or fully implemented. (user administration, viewing of relationship of the objects, domain-functional web-gateway, handling of independent separate links, etc). More detail and negative aspects of this choice will be discussed in the next chapter.

The central database management system was represented and powered by Hyperwave with its containing object-oriented database (OB). Its' separation of the server into an object server (OB), document server (DC) and full text server (FT) enables very fast searching of the indexes, i.e. the metadata. It can be run in a server pool, enabling multi-server / multi client environments in WAN with its own proprietary hyperG protocol. Its separate protocol conversion layer enables the use of different and additional protocols, as also the http protocol with a web-gateway. The Hyperwave server with a web-gateway was installed on a Silicon Graphics Indy running Irix 5.3.

An external simplified Z39.50 -gateway was linked to it by a perl script, transposing incoming requests into the object query language used by Hyperwave. The hyper-G protocol allows proprietary clients (as for instance Harmony and Mozart) to have session-based protocols with graphical browsing abilities. To a certain extent, these graphical representations of relationships of the objects in the DBMS can also be realised through the http protocol with Java implementations.



The system can run simultaneously with a normal webserver on the same machine. Whereas the main web pages of the SMaTBaM prototype depicting the content of the database can be accessible through one port (here http://scream:8083), proprietary clients can also connect using the proprietary session-based protocol hyperg (here hyperg://scream:8084). It is possible to run several different web-gateways with different presentation schemes on different ports. Also single objects can be linked to certain www-gateway presentation modules, and single presentation modules can be programmed with a c++-like programming language. Thus there is a scalable control down to object level of the object presentation over the WWW and it is possible to have different views on one object.

The maintenance and user administration is also possible through the www-gateway through a different port. (Here scream:8084). As the media server we used MediaBase on a Silicon Graphics Challenge (here soundserver:5049). Movie players or soundplayer, proprietary of MediaBase or of the University of Glasgow were able to make direct connections over the ATM networks. These players were installed on the client machines as browser plugins. Sound and movie objects were than put into hyperwave as objects, calling the functions by scripts to establish the direct ATM connection either through Java classes or C++ classes.

(For a demonstration of the aspects of the SMaTBaM prototype system, see

"*Workshop SMaTBaM, (Friday 13, June 1997), slide 07*" at
http://www.music.gla.ac.uk/HTMLFolder/Research/smatbam-private/smatbamWorkshopNo1/slide07.html.)

7.4. Aspects of Database Schema Design

The design of the database content will be influenced to some extent by several factors:

- the work of the service provider's collection officers themselves
  Using an object-oriented design for mapping knowledge domains into database schemas will improve the searching and browsing capabilities of users accessing items. One of the main problems in this domain is, what kind of objects exist in the area of the performing arts, or what kinds of categories and what kind of functionality / behaviour do they have. Additionally, what kind of relationships exists between them. These relationships can be implemented as either
  - associations between objects/instances or
  - as attributes which store relationships in a main class "Collection Item"
  - or as hyperlink references not stored in the items themselves but rather separately as own objects
- the database chosen
  Even though OOA and OOD in theory should be independent of the implementation language used, realistically it does make a difference if for instance a C++ OODBMS or a Smalltalk or a Lisp OODBMS is used. Characteristics as dynamic typing or multiple inheritance influences the design.
- the functionality of the web-gateway
  The next chapter will show, that the main differences of the vendors evaluated in the context of searching an efficient system for a SMaTBaM service was not performance, but rather functionality. The performance requirements of a SMaTBaM service are rather low on the object server side when compared to telecommunication uses of OODBMS. Objects stored in the context of digital resource libraries will not likely overstep a certain level of complexity or compositeness. Additionally, if http is used as a protocol, the high performance of a OODBMS will never be reached in a WAN environment due to the limitations of the stateless http protocols. Performance issues as for instance the performance of serving BLOBs are discussed in the next chapter. In general, the performance issues lies more in the delivery of the objects, i.e. the web-gateway and in the presentation of the objects in a WAN context, i.e. the functionality of the web-gateway.

In the SMaTBaM prototype system using Hyperwave and MediaBase, relationships between objects are shown by either as

- separately stored links between objects (links are bidirectional and are allowed to reach into video / sound / image objects, as well as into texts)
- references to other mime-type objects on the web or database, but in the object/document itself
- certain defined relationships as sequence, cluster, annotations, etc.

Additional functionality is planned to be implemented by expanding HyperG's own system object hierarchy. (see http://scream.music.gla.ac.uk:8083/0x811b9908_0x00242ffd.) As for instance by adding classes or attributes into the Object architecture of the Hyperwave OODBMS, expanding the HyperG attribute set of the Class "Document" with MetaDataAtrtibuteSet, as can be seen below.

Creation of a new object "CollectionItem" with at least the following attributes

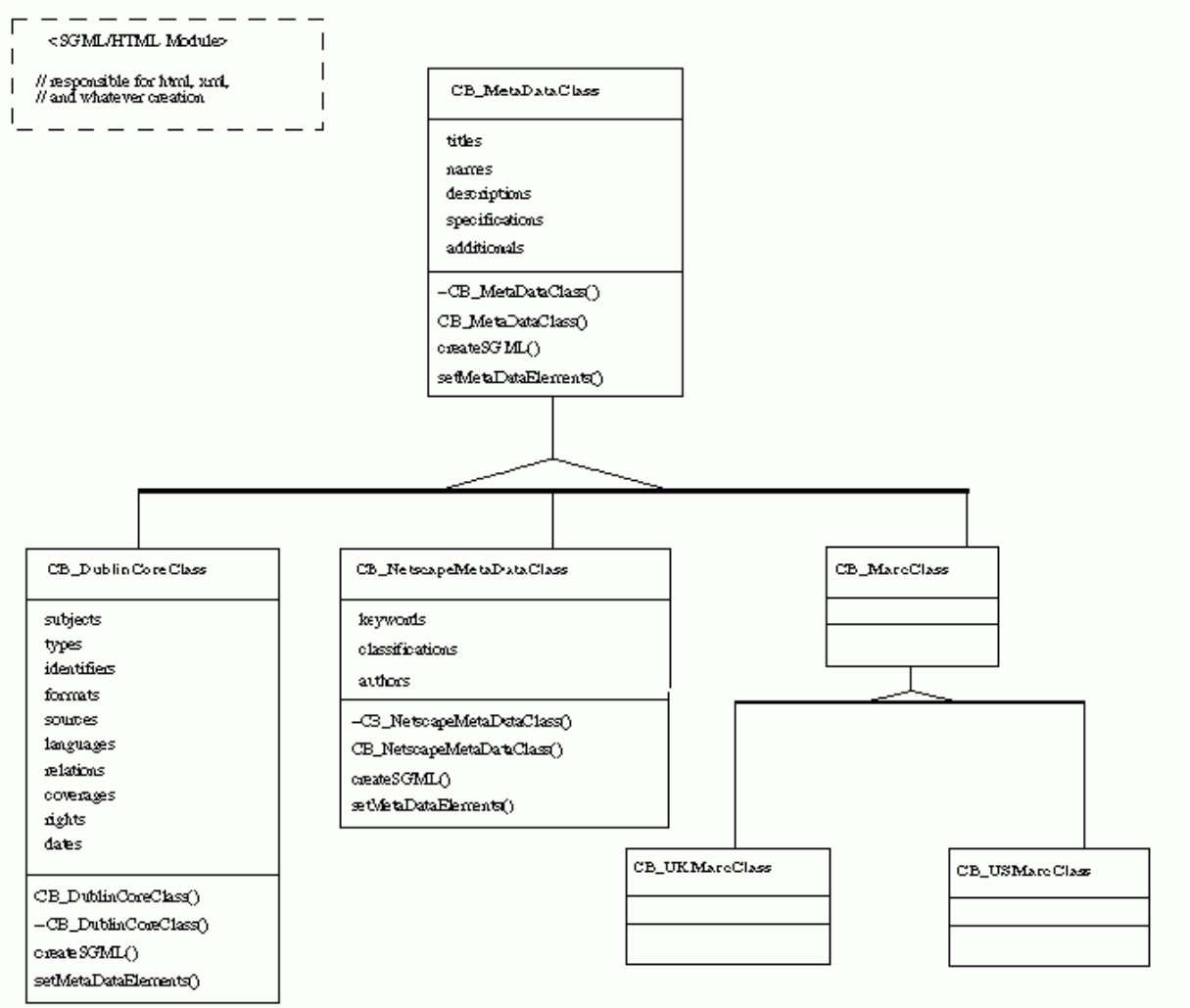
Class: CollectionItemClass

Attributes:

- metadata
- goid
- oid
- relationships (see below)

Whereas the attribute metadata is again of the class MetaDataClass:

Class: CB_MetaDataClass



and the DublinCoreClass attributes are either a record of certain discipline specific subtypes or a of the type of a class defined for these subtypes. (Difference do to the availability of defining attributes as instances of a class or, if that is not possible, as basic language data types).

Browsing objects in a object-oriented databases mainly is realised nowadays by three different web-gateway functionalities:

- a) the web-gateway functions mainly as a forms-interpretation module queering the database content through text-index searches, and the results are depicted straightforward as structured and linked texts or / and images (most commercial web-gateways stand under this category)

- b) the web-gateway handles the attributes of objects in a manner of transforming them into a "browsable" graphic presentations of objects in relationship with each other on a WAN-client. They might be browsable either through text, 2-dimensional graphics, or 3-dimensional landscapes of objects. (this case is with the choice of the SMaTBaM prototype system, using Hyperwave and MediaBase in addition to some additional classes and scripts)

- c) the relationships between the objects themselves are depicted using OQL or ODL, being rather a interactive "Case-tool" representing the objects according to a certain OOD-method. (Booch, Rumbaugh, etc.) (As far as I know, there is no commercial system around of this functionality, although there are some research papers published.[36]

Main difficulty with all three categories of web-gateways is the unstatefulness of the http-protocol, which makes normal database session almost impossible. Solutions have been implemented with either using cookies on the client side, calling CGI's, or calling session-based protocols from the WAN client. There have been also research works in using intermediate server side dispatcher scripts, responsible for keeping the connection to the OODBMS open and storing needed text data for representing the objects on WAN-clients. [37]

If it was not the general acceptance of the http protocol as the main protocol for interoperability on WANs, one Q.E.D. of this work would be: Efficient and acceptable browsing of a database content can only be achieved through the use of session-based protocols.

In all three examples of browsing objects in a OODBMS, the "meaning" of the relationship between objects/documents is vital for an efficient searching mechanism. Thus one of the main tasks of a information service utilising an OODBMS would be to design and implement a schema, which defines different types of relationships existent between objects existent in the context of a digital resource library. The relationships will be analogue to a in-depth Metadata index known from the conventional library. Thus we have relationships as for instance in this simple music example:

- PersonB composed-by WorkA
- WorkA contains WorkA1, WorkA2, WorkA3
- WorkA is TypeA
- WorkA can-be-played-by INstrumentA or / and InstrumentB
- PersonB lived-in PLaceA
- PlaceA is-contained-by PLaceB
- WorkA is-owned-by PersonC
- etc

A flexible and efficient way of mapping information into objects/documents with specific characteristic relations will enable the user to access and search for objects in its inherent context. Thus the utilisation of a database schema defining these different kinds of relationships between knowledge domains in the performing arts will be of high relevancy. The characteristics of the relationships should be expandable and flexible. Thus a general class Relationship should be defined, its instances being certain specific relationships between objects/documents.

Class Relationship inherits down to Class RelationshipByReference and RelationshipByContainment.

Class Hierarchy

- Class Relationship
    - Class RelationshipByReference
    - RelationshipByContainment

---

[36] C. Varela, D. Nekhayev, P. Chandrasekharan, C. Krishnan, V. Govindan, D. Modgil, S. Siddiqui, D. Lebedenko, M. Winslett, DB: Browsing Object-Oriented Databases over the Web, Proceedings of the fourth international WWW Conference, Massachusets, December 1996.
http://www.w3.org/Journal/1/varela.282/paper/282.html
and Sjolin M. A WWW Front End to an OODBMS. The Second International Conference on the World Wide Web, Oct 17-21, 1994, Chicago, Illinois, U.S.A.
http://www.ncsa.uiuc.edu/SDG/IT94/Proceedings/Databases/sjolin/sjolin.html

[37] ibid.

with

Class Relationship

attributes:

- ObjectsRelatedTo
- ObjectRelatedFrom
- RelationShipType

methods:

- getObjectsRelatedTo (anElement:resourceObject)
- getObjectRelatedFrom (anElement:resourceObject)
- getRelationShipType (anElement:resourceObject)
- setObjectsRelatedTo (anElement:resourceObject)
- setObjectRelatedFrom (anElement:resourceObject)
- setRelationShipType (anElement:resourceObject)
- createRelationShipTypesOurOfMetaData (aMetaDataSet:MetaData)
- createMetaDataOutOfRelationships (anElement:resourceObject)

Class RelationShipView

Creates a graph out of the relationships of a object.

A similar architecture was defined but not yet standardised yet in the Corba Service specifications, written in 1995 and formalised in 1997 [38]. It might be worthwhile thinking an implementation using the CORBA architecture in utilising these implemented relationship classes for the digital library.

These example classes can be seen as just some realisation of aspects in designing a system for the use in giving users a context-based method of access and searching in digital resource libraries. The PADS project will hopefully demonstrate the capability of context-based searching on a WAN with the aid of object-oriented technologies.

8. Evaluation and Results

8.1. The evaluation process of required hard- and software

8.1.1. General information about the evaluation process

From the projects' first general operational requirements and user specifications, it was clear that a SMaTBaM prototype system was supposed to fulfill several tasks at once. To find a system aimed perfectly in its functionality, as in its performance in such specific but broad range target area was highly unlikely. A major exterior issue was also the use of this system in the PADS project, which in its objectives does not include any R&D tasks for designing and implementing exactly such a system.

For the SMaTBaM project, which ran in the timespan of 9 months from October 1st to July 31st, this meant to target evaluation and implementation results which will be of practical use to the PADS service after the close of the SMaTBaM project:

---

[38] CORBAservices - Relationships Service, OMG, ftp://ftp.omg.org/pub/docs/formal/97-07-18.ps

- set up a system which can be practically used immediately
    - with the ability of streaming audio and video over ATM networks
    - with access methods relevant for a SMaTBaM service and as described in this previous chapters, such as
        - logical search methods
        - controlled access
        - graphical representations of the DBMS content
        - compliance to evolving digital library standards
        - etc.

It was already clear form the specification of the future content material of a SMaTBaM service, that systems deriving form the OODBMS or ORDBMS world should be intensively looked at. The material being of highly interrelated information and its inherent complexity calls for handling it in an object-oriented fashion.

From the delivery mechanisms it was clear that a restricted and controlled access over WAN(/MAN) was highly wished. Also new ways of searching and representing data on WAN was targeted. In this area web-gateway products and past research results of the hypermedia world came up with solutions which were thought over for deployment in a SMaTBaM-like service.

In the area of streaming BLOBs over the high-performance networks, specialised media-server hard- and software was researched as an efficient, scalable, flexible and fast solution.

The basic architecture of using a OODBMS and a web-gateway to manage WAN functionality was logical from the start. To achieve the most efficient point of start, combination of products were looked at from the areas mentioned above, as well as the few existent one-vendor solutions. With the aim to keep the implementation effort in a short-terms as well as the as long-terms to an acceptable amount of implementation work.

The whole process did show, that main emphases in the evaluation process for a SMaTBaM-like service had to be put into functional comparison, rather than performance issues. Main implementation strategies for the prototype as for the long-phase plans for the system are targeted rather in the expansion of an existing system, the establishing of gateways between systems, than the whole new design from zero up to maximum functionality.

8.1.2. Evaluation Criteria

There are quite a few lists of evaluation criteria for OODBMS around, exhausting any person researching into evaluation processes. One of the most detailed ones is Douglas Barries' "ODBMS Feature Checklist" containing 184 evaluation criteria subgrouped into 13 Categories with a total of ca 1300 features to be checked.[39] Another evaluation list with 72 evaluation criteria but subgrouped into only 3 basic categories of functionality, application development issues and miscellaneous criteria can be found[40] .

Main criteria that were found to be of relevancy for the performance and the functionality of a SMaTBaM-like system were

Database Management system

- object-oriented or object-based database management system

---

[39] Barry, Douglas K., "ODBMS Feature Checklist." Object Magazine, (January-February 1993) also available at
http://www.odbmsfacts.com/

[40] Gregory McFarland, Andres Rudmik, Object-Oriented Database Management Systems, A Critical Review/Technology Assessment, Data & Analysis Center for Software (DACS), Department of Defense (DoD), Information Analysis Center (IAC), September 1993
http://www.utica.kaman.com/techs/OODBMS/oodbms.ToC.html

- high support or delivery, maintenance and manipulation of multimedia data types
- support of BLOB storage

Web-gateway

- graphical representation of the DBMS content
- integrated session-based functionality
- possibility of setting up a controlled intranet on a WAN basis
- database content representation schemes
- modular protocol-layer architecture

Streaming

- scalable rate transmission
- ATM support

None of these evaluation lists mentioned above take these additional functionality extending the central OODBMS into consideration. For the SMaTBaM proposed system, an own derived list of evaluation criteria was developed. Some functionality might not be met by the system as being present after installation. The system should show a possibility of implementation in medium time-spans. In general, the basic system architecture should contain a ODBMS, a web-gateway and a means of streaming audio and video.

8.1.2.1. Web-gateway Functionality

- Hypermedia language support
    - Supports html/sgml/xml/vrml page creation
    - Supports multi-media objects in the delvery process
      Describe in which way hypertext is generated and in which amount it is configurable
      Describe if and how certain hypermedia techniques are supported, as for instance links into video or sound data
- Supports use of different WAN protocols
    - http, CORBA, cgi, tcp/ip, atm
      list of realised and planned protocols
      describe implementation effort for realising relevant protocols
- Supports (graphical) representation of object/document relationships
    - is-a , has-a , is-associated-with
    - other
      describe if and how data is represented graphical on the user side
- Supports some kind of use of session-based protocols for WAN-DBMS access
    - through CGI
    - through own protocols
    - etc
      description
- Supports controlled access to objects/documents in the ODBMS
    - Supports hierarchical user groups
    - Object/document level access rights
      descriptions

- Distributed objects
    - Support of standards for using distributed object technology
      description of how distributed objects are supported
- Object/document Identity
    - GOID
    - OID
      Describe the size and format and object identifiers.
      Describe how object identifiers are used to identify a specific object.
- Objects/documents can have arbitrary attributes
    - Supports internet data types, indexing of attributes, standard library metadata schemes
      Describe how arbitrary types can be utilised for the storage of documents and how the attributes can be presented on the client
      Implementation effort if not available
- Objects/document versioning?
- Locking over the web-gateway?
- Supports composite objects over the web-gateway?
  Can the composite objects be handles as one, as for instance in copy, delete actions?
- Integration with Existing DBs and Applications
  Description
- Distributed Client - Server Approach
    - Supports distributed access to data.
    - Uses a client-server architecture.
    - *Describe the overall architecture of System including all interacting processes (e.g., client, server, lock, etc.)*
    - *Describe mechanisms (if present) used to manage secondary storage (e.g., as files, as disk partitions, etc.)*
- Data Access Mechanism
    - *Describe how the system transfers data between servers and clients(OODBMS performance will be closely tied to the data access mechanism)*

8.1.2.2. Database Functionality

- Object-oriented modelling criteria
    - Complex and composite objects
      Supports definition of complex and composite objects
    - Object identifiers
        - GOID
        - OID
          Description of ID
    - Support of Multimedia objects and BLOBs
      Describe multimedia classes and their functionality, BLOB support
    - Supports dynamic or static typing (of attributes and/or objects)
    - Multiple/single inheritance

        - Location transparency
        - Persistency
          Describe how persistency is achieved
          Describe how persistent objects are accessed from within an application. In particular, are they referenced via pointer or by some other mechanism? Also, are they implicitly loaded and stored from the database or must data access be explicitly programmed? Finally, must the application explicitly identify updates to objects (so that the OODBMS knows what objects need be write locked and copied back to the database)?

- Database Architecture
    - Client/Server and Storage architecture
        - multiple client / multiple server
        - multiple client / single server
          Description if needed
    - Platforms Supported
      list platforms supported for all elements of the system, and if heterogenous distribution of database elements is possible (database server, streaming media server, web-gateway server)
    - Cache Location
      describe location (client or server) and size / tuning of cache
    - Concurrency
      Describe the concurrency control policies provided by the OODBMS and how an application's access to data is affected by these policies.
    - Locking and Transaction
        - Supports object level locking.
        - Supports page level locking.
          Describe the transaction processing and associated overheads. Describe the use of locks for transaction implementation, especially with regard to how and when objects are locked and when locks are released.

- Database administration
    - On-line maintenance
        - Supports backup.
        - Backup can be performed while the database is being accessed.
        - Backups can be performed on specific segments of the database.
        - Reconfiguration

- Query language
  list
- Indexing
  Description
- Application Programming Interface
    - ODL/DDL/DML Language
    - C++, Java, Smalltalk,
- External Vendor Support
  Description and list

- Standards
    - interface to which ST, C++, Java programming environments, compilers?
    - Member of or compliant to ODMG, OMG
    - compliant to ORB, ODBC, OLE
    - Product uses standard version of a programming language.
      Description

8.1.2.3. Application inherent criteria (system design for a service such as in the projects SMaTBaM, PADS and MusicWeb)

- general remarks (pros and cons) in the context of a SMaTBaM service
    - Websupport
    - Multimedia support
    - bottlenecks
    - SMaTBaM solutions to serving massive time-based media
    - high performance techniques
      Description
- Application Development Process
  Description of steps in developing an application including schema design, application development, coding, testing, and debugging
- Application Development Tools
    - Database Administration Tools
    - Database Design Tools
    - Database Browsing Tools
    - Special application support classes

8.1.2.4. Miscellaneous Criteria

- Product Maturity
  Describe the product maturity based on criteria as
  years under development, number and type of utilisations of the product
- Product Documentation
- Vendor Maturity
  Describe the company's maturity in terms of size, age, staff, and financial stability.
- Vendor Training
  Describe the training classes offered for application developers and database administrators.
- Vendor Support & Consultation
  Describe the support
- Vendor Participation in Standards Activities and Research projects
  Describe the standards activities the vendor participates in (OMG, ODMG, ANSI , ISO, etc)

8.2. Designing specific problem inherent Benchmarks

As mentioned in the above chapters, a more relevant issue in designing and deciding on a specific system architecture with one or a combination of vendor products, was the functionality rather than the performance. In a benchmark, when trying to measure the performance of a system or a combination of systems in its "unended form" a well-defined task or set of tasks should be defined, which are able to "predict the performance of an unknown system on a known, or at

least well-defined, task or workload"[41].

Tasks or scenarios which can be identifies in a SMaTBaM system of the above give system requirements is

Remote vs local access

1. Access to massive time-based media over a WAN
    1. access time and quality of delivery of a single BLOB by a single user over ethernet
    2. access time of a quality of delivery single BLOB by a single user over ATM
    3. access time of a quality of delivery single BLOB by a multiple users over ethernet
    4. access time of a quality of delivery single BLOB by a multiple users over ATM
    5. access time of a quality of delivery different BLOBs by a multiple users over ethernet
    6. access time of a quality of delivery single BLOB by a multiple users over ATM
2. Access to a composite object
    1. access time of a composite object by a single user over ethernet
    2. access time of a composite object by a single user over ATM

Number of Users

1. Multiple access to one object
2. number of maximum users accessing the database
3. number of maximum sessions of streaming audio and video
4. number of maximum users allowed for controlled access

Access comparison between warm and cold time

1. Queries queries: system attributes, arbitrary attributes
    1. timespan for results querying locally
    2. timespan for results querying over the web-gateway locally
    3. timespan for results querying over the web-gateway over ethernet on a weekday
2. text search: in arbitrary attributes, full-text search
    1. timespan for results querying locally
    2. timespan for results querying over the web-gateway locally
    3. timespan for results querying over the web-gateway over ethernet on a weekday

8.3. Results

Results are published as **internal documents**. For anybody wishing to receive a copy of the results, email me (Carola Boehm) or Stephen Malloch. The internal documents can be accessed by user name and password. A word of thanks to all the vendors providing us with support in form of discussion, ideas, solutions and evaluation licenses.

10. Bibliography

Information about DBMS and Benchmarks

General Information on OODBMS and the evaluation process

---

[41] Benchmarks FAQ Version 0.6, Tue Mar 28 22:57:29 1995
http://hpwww.epfl.ch/bench/bench.FAQ.html

- The object-oriented database Manifesto [Atkinson+deWitt+alii 89]
  The golden rules , a classic paper defining OODBMS and describing the main features and characteristics that a system must have to qualify as an OODBMS. The beginning of OODBMS theory and praxis. Most of the authors of this paper started their own OODBMS company.

- Objektdatenbanken fuer kommmerzielle Client/Server Anwendungen [Micram95]
  (ODBMS for commercial client/server applications)
  The paper provodes an overview of different OODBMS architectures and theyr consequences for perfromance and target application areas. Several main target application areas are described in addition to their performance and functional requirements. Micramis a distributor of Objectivities' ODBMS.

- Objectivity White Papers: Choosing an ODBMS [Objectivity97]
  General paper about choosing a object-oriented database, written by Objectivity.

- OBJECT-ORIENTED DATABASE MANAGEMENT SYSTEMS.A Critical Review/Technology Assessment[DACS97]
  This report reviews the state of the art of Object-Oriented Database Management Systems (OODBMS). The objective of this report is to provide the reader with an understanding of the issues relevant to OODBMS technology and to describe where commercial products stand on these issues. (ODBMS: Versant, Objectivity, Ontos, GemStone) This report can be used as the first step in an evaluation aimed at selecting an OODBMS for use in a given application development effort. It describes a broad range of OODBMS evaluation criteria. It then evaluates five commercial OODBMS products on a subset of the overall criteria.

- The Evolution of Client/Server Architectures [SPL WorldGroup97]
  This paper represents an easy to understand introduction to the evolution from 2-tier to 3-tier to multi-tier client/server applications and puts commercial databases in this context. SPL World group is a comapny specialising in client/server services.

- Why Use an ODBMS? A Comparison Between Relational and Object Oriented Databases for Object Oriented Application Development [POET96]
  This White Paper is intended to clarify the differences between relational and object oriented database systems, especially from the perspective of object oriented application development, using programming languages like C++ or Smalltalk. A fine difference in this paper written by POET should be noted in their use of ODBMS and not OODBMS.

- Welches OO Datenbanksystem ist das richtige? Leitfaden zur Auswahl eines ODBMS. (Which OODBMS? A guide for choosing OODBMS')
  A general paper describing methods and the process of evaluating and choosing an ODBMS. The author is from Siemens, Muenchen and is not affiliated to any OODBMS vendor.

Benchmarks and Comparisons

- Object fault handling of persistent programmin g languages: A performance evaluation [Hosking93]
  The paper represents a number of mechanisms for detecting and handling references to

persistent objects, and evaluates their relative performance within an implementation of Persistent Smalltalk

- The 007 Benchmarks [Carey+DeWitt+alii94]
  The 007 benchmarks represent a comprehensive test of OODBMS performance. Description of the benchmark and performance results. The benchmarks were originally designed to simulate CAD/CAM environments.

- A study of three alternative workstation-server architectures for object oriented systems [deWitt+Futtersack]
  A report analyzing the performance of three different OODBMS architectures: object server. page server and file server.

- A taxonomy for secure object-oriented databases [Olivier+Solms1994]
  This paper proposes a taxonomy for secure object-oriented databases in order to clarify the issues in modelling and implementing such databases. It also indicates some implications of the various choices one may make when designing such a database.

- The BEAST Benchmark [BEAST]
  he BEAST Benchmark The Beast Benchmark Beast is a benchmark dedicated to (object-oriented) active database management systems. It uses the schema and databases of the OO7 benchmark .This paper presents the first comparative performance study of object-oriented active database management systems by using the BEAST benchmark. (ODBMS: ACOOD, Ode, REACH, and SAMOS)

- Fine-Grained Sharing in a Page Server OODBMS [Carey+Franklin+Zaharioudakis1994]
  Several approaches are described, including an adaptive granularity approach that uses page-level locking for most pages but switches to object-level locking when finer-grained sharing is demanded.. A study of the performance of these approaches, comparing them to both a pure page server and a pure object server.

- OBJECT-ORIENTED DATABASE MANAGEMENT SYSTEMS.A Critical Review/Technology Assessment[DACS97]
  This report reviews the state of the art of Object-Oriented Database Management Systems (OODBMS). The objective of this report is to provide the reader with an understanding of the issues relevant to OODBMS technology and to describe where commercial products stand on these issues. (ODBMS: Versant, Objectivity, Ontos, GemStone) This report can be used as the first step in an evaluation aimed at selecting an OODBMS for use in a given application development effort. It describes a broad range of OODBMS evaluation criteria. It then evaluates five commercial OODBMS products on a subset of the overall criteria.

SMaTBaM and OODBMS related information material

- Paradigms and Environments for the Development of Distributed Realtime Systems , [Nielsen94]
  This Phd project focuses on new implementation paradigms and environments for the development of software for distributed real-time systems.

- Integrating Object Technology and the Web [W3C96]
  This is an investigation of the integration of distributed object technologies with the web,

and the duality between application programmer interfaces (APIs) and network protocols.

- Distributed Objects on the WWW: A Position Paper [Resnick1996]
  This paper explores some recent initiatives such as the ORB-to-Java gateways, Apple's CyberDog, Sun's JavaBeans and IBM's Arabica in integrating CORBA, OpenDoc and Java. It then considers how to go beyond these attempts into a domain where the full benefits of the WWW, Java, CORBA and OpenDoc are fully exploited in a seamless whole.

- Z39.50 Profile for Access to Digital Collections, Draft seven [Loc96]
  Profile specifies a conforming subset of the Z39.50-1995 and addresses problems for access to digital collections. Provides semantics for navigating digital collections, to locate and retrieve objects of interest.

- Providing Social Interaction in the Digital Library [Ackermann94]
  This paper discusses social interaction possibilities in digital libraries, and describes an implementation (Cafe ConstructionKit) providing computer-mediated communication support at low cost.

- Ein OO-Framework fuer Netzwerkprotokolle, [Hueni+Keller97]
  German paper about presenting a design and implementation of using frameworks in the implementation (C++) of ATM protocols.

- A framework for network protocol software [Hueni+Keller97]
  Using Conduit+, a framework for designing and implementing network software with a componentized object-oriented framework and object-oriented design patterns.

- Object-Orineted Components for High-Speed Network Programming, [Schmidt+Harrison+alii95]
  This paper makes two contributions to the development and evaluation of object-oriented communication software. It reports performance results from benchmarking several network programming mechanisms (such as sockets and CORBA) on Ethernet and ATM networks. The paper describes the software architecture and design principles of the ACE object-oriented network programming components.

- DB: Browsing Object-Oriented Databases over the Web, [Varela+Nekjayev+alii96]
  In this paper, issues are presented that arise when users browse object-oriented databases over the World Wide Web, and their performance results for the Database Browser (DB) implementation. (problems http, statefullness with CGI scripting, ODL for schema browsing)

- A WWW Front End to an OODBMS. [Sjolin94]
  The authors have created a prototype World Wide Web (WWW) front end to an OODBMS, LINCKS. LINCKS is a multi user object store with support for composite objects, database histories, multiple views, information sharing by linking, and parallel editing notification.

---

11. References

[Ackermann94] Mark S. Ackerman, Providing Social Interaction in the Digital Library, Published in

Proceedings of Digital Libraries '94: First Annual Conference on the Theory and Practice of Digital Libraries (College Station, TX), pp. 198-200 . http://www.ics.uci.edu/~ackerman/docs/dl94/dl94.final.html

[Atkinson+deWitt+alii 89] M. Atkinson, F. Bancilhon, D. DeWitt, K. Dittrich, D. Maier, and S. Zdonik. The Object-Oriented Database System Manifesto. In Proceedings of the First International Conference on Deductive and Object-Oriented Databases, pages 223-40, Kyoto, Japan, December 1989. Also appears in Building an Object-Oriented Database System: The Story of O2. F. Bancilhon, C. Delobel, and P. Kanellakis. (eds.) Morgan Kaufmann, 1992.http://www.cs.cmu.edu/People/clamen/OODBMS/Manifesto/index.html

[BEAST]Andreas Geppert, Mikael Berndtsson, Daniel Lieuwen, Jürgen Zimmermann, Performance Evaluation of Active Database Management Systems Using the BEAST Benchmark, Insitut fuer Informaktik, Universitaet Zuerich, The BEAST Benchmark. http://www.ifi.unizh.ch/groups/dbtg/Beast/

[Carey+Franklin+Zaharioudakis1994] Michael J. Carey, Michael J. Franklin and Markos Zaharioudakis, Fine-Grained Sharing in a Page Server OODBMS, UW Madison Computer Sciences Technical Reports, CS-TR-94-1224, April 1994.
http://www.cs.wisc.edu:80/Dienst/UI/2.0/Describe/ncstrl.uwmadison/CS-TR-94-1224

[Carey+DeWitt+alii94] Michael J. Carey, David deWitt, Jeffrey F. Naughton, The 007 Benchmark, Computer Science Department, Universoty of Wisconsin Madison, Wisocnsin 1994.

[DACS97] Gregory McFarland, Andres Rudmik, OBJECT-ORIENTED DATABASE MANAGEMENT SYSTEMS, A Critical Review/Technology Assessment,The Data & Analysis Center for Software (DACS), Department of Defense (DoD) Information Analysis Center (IAC), September 1993
http://www.dacs.dtic.mil/tech/oodbms.ToC.html

[deWitt+Futtersack] David. J. DeWitt, Phillippe Futtersack, David Maier, Fernando Velez, A study of three alternative workstation-server architectures for object oriented systems , Comp uter science Department of the University of Wisconsin 1990. ftp://ftp.cs.wisc.edu/tech-reports/reports/90/tr936.ps.Z

[Hosking93] Antony L. Hosking, J. Eliot B. Moss, Object fault handling of persistant programming languages: A performance evaluation, Proceedings ACM Conference on Object-Oriented Programming Systems, Languages and Applications. Washington, DC, Sep.1993, pp.288-303.
ftp://ibis.cs.umass.edu/pub/papers/oopsla93.ps.Z

[Hueni+Keller97] Hermann Hueni, Beat Keller, Ein OO-Framework fuer Netzwerkprotokolle, Objektspektrum Freburar 1997, p.51-57.

[Hueni+Johnson+alii95] Hermann Hueni, R. Johnson, R. Engel, A framework for network protocol software, in: Proceedings of OOPSLA-95, 1995.
ftp://st.cs.uiuc.edu/pub/patterns/papers/conduits+.ps

[Loc96] Library of Congress, Z39.50 Profile for Access to Digital Collections, Draft seven, May 3, 1996.

[Micram95]Mathias Frei, Marc Gille, Objektdatenbanken fuer kommmerzielle Client/Server Anwendungen (ODBMS for commercial client/server applications), Conference paper of SoftwareDevCon 1995, Wiesbaden 1995.

[Nielsen94] Brian Nielsen, Paradigms and Environments for the Development of Distributed Realtime Systems, 1994.
http://www.iesd.auc.dk/~bnielsen/Phd/phdproject.html

[Objectivity97] An Objectivity, Inc. White Paper, CHOOSING AN OBJECT DATABASE, 1997
http://www.objy.com/ObjectDatabase/WP/Choosing/Choosing.html

[Olivier+Solms1994] Martin S. Olivier, Sebastiaan H. von Solms, A taxonomy for secure object-oriented databases, ACM Transactions on Database Systems, Vol. 19, No. 1 (March 1994), Pages 3-46.
http://www.acm.org/pubs/citations/journals/tods/1994-19-1/p3-olivier/

[POET97] Poet Software, Why Use an ODBMS? A Comparison Between Relational and Object Oriented Databases for Object Oriented Application Development.
http://www.poet.com/t_oovsre.htm

[Resnick96] Ron I. Resnick , Distributed Objects on the WWW: A Position Paper, 1996.
http://www.interlog.com/~resnick/position.html

[Schmidt+Harrison+alii95] Dourglas Schmidt, Tim Harrison, Ehab Al-Schaer, Object-Orineted Components for High-Speed Network Programming, USENIX Conference on Object-Oriented Technologies, Monterey, CA, June 1995.
http://www.usenix.org/publications/library/proceedings/coots95/schmidt.html

[Sjolin94] Sjolin M. A WWW Front End to an OODBMS. The Second International Conference on the World Wide Web, Oct 17-21, 1994, Chicago, Illinois, U.S.A.
http://www.ncsa.uiuc.edu/SDG/IT94/Proceedings/Databases/sjolin/sjolin.html

[SPL WorldGroup] Reading SPL WorldGroup, The Evolution of Client/Server Architectures, 1997.
http://www.splwg.com/ClientServer_Evolution.html

[Varela+Nekjayev+alii96] C. Varela, D. Nekhayev, P. Chandrasekharan, C. Krishnan, V. Govindan, D. Modgil, S. Siddiqui, D. Lebedenko, M. Winslett, DB: Browsing Object-Oriented Databases over the Web, Proceedings of the fourth international WWW Conference, Massachusets, December 1996.
http://www.w3.org/Journal/1/varela.282/paper/282.html

[W3C96] Integrating Object Technology and the Web.1996
http://www.w3.org/pub/WWW/OOP/

Appendix A

Contact Persons and Sites of Importance

Contact Persons

mailto:

Stephen Arnold, S.Arnold@music.gla.ac.uk

Carola Boehm, carola@music.gla.ac.uk
Celia Duffy, celia@music.gla.ac.uk
Stephen Malloch, mallocsc@music.gla.ac.uk
Catherine Owen, cowen@music.gla.ac.uk

Sites

- SMaTBaM Welcome Page
- PADS
- PADS Hypermail Archives
- AHDS

- Web Gateways to SMaTBaM
  - Web-gateway to HyperG / normal
  - Web-gateway to HyperG / Database Maintenance
  - Web-gateway to VisualWave
  - Z39.50 Isite Search Page to SMaTBaM Collection (not working yet)

- The Z39.50 Protocol
  - The ANSI/NISO Z39.50 Protocol
  - Z39.50 maintenance agency home page
  - Z39.50/SQL+ - Stateful Web Access to Relational Databases (5th July 1996)
    http://www.dstc.edu.au/DBU/research_news/papers/zposition.html
    A proposal for the extension of the Z39.50 standard to include SQL access for information retrieval. Submitted for the agenda for discussion at ZIG meeting in October,1996.
  - Z39.50/SQL+ - Proposal for SQL Access in Z39.50 (19th April 1996)
    http://www.dstc.edu.au/DDU/research_news/reports/zproposal.html
    A Positioning statement regarding the Z39.50/SQL+ project submitted to the Distributed. Indexing/Searching Workshop 1996
  - Z39.50/SQL+ Project (27th February 1996)
    http://www.dstc.edu.au/DDU/research_news/reports/zreport.ps
    A Technical Report outlining the theoretical and implementation issues of the Z39.50/SQL+ project
  - NEXT ZIG MEETINGS (Z39.50 Implementors Group (ZIG))
    (http://lcweb.loc.gov/z3950/agency/zig/zig.html)

- The Dublin Core and Metadata
    - Dublin Core Metadata Template
    - http://www.roads.lut.ac.uk/Metadata/DC-ObjectTypes.html
    - Dublin Core Metadata Element Set: Resource Page
    - Dublin Core Metadata Element Set: Reference Description

- The Warwick Framework
    - The Warwick Framework
    - MIME Implementation for the Warwick Framework

- Related Projects
    - AHDS Newsletter Index
    - AHDS. Internal Working Documents
    - AHDS: http://www.kcl.ac.uk/projects/ahds/top.html
    - MIniMS: http://www.dcs.gla.ac.uk/~george/minims/minims.html
    - NetMuse: http://www.music.gla.ac.uk/HTMLFolder/Research/NetMuse.html
    - S.C.R.A.N. Info
    - Site of the MusicWeb project
    - ADAM System Statement of Requirements

- Current projects with similar objectives and scope
    - LIBERATION
    - Frquence Banane
    - S.C.R.A.N. Info
    - IICM's Public Services
    - D-Lib Magazine
    - LDAP Frequently Asked Questions

- Related Information Sites
    - SHEFC: http://www.shefc.ac.uk/shefc/welcome.htm
    - GU TILT: http://www.elec.gla.ac.uk/TILT/TILT.html
    - TLTP: http://www.icbl.hw.ac.uk/tltp
    - ATM: http://www.icbl.hw.ac.uk/ltdi/man-info/atm.html
    - JISC: http://www.niss.ac.uk/education/jisc/
    - EPSRC: http://www.epsrc.ac.uk/

        - EPSRC Multimedia and Networking Applications Programme: http://www.epsrc.ac.uk/progs/area/it_cs/mna-cont.htm
        - InterNIC Internet Documentation (RFC's, FYI's, etc.)
        - EC - The Educational Multimedia Task Force: Related WWW sites
        - EC - Guide to tasks covered by Esprit calls in 1997
- Object-oriented Information Sites
    - Standards
        - W3C - The World Wide Web Consortium
        - Object Database Management Group Home Page
    - Benchmarks and Comparisons
        - ftp://ibis.cs.umass.edu/pub/papers/oopsla93.ps.Z
        - ftp://ftp.cs.wisc.edu/tech-reports/reports/90/tr936.ps.Z
        - http://www-db.att.com/acquire.html http://www-db.att.com/acquire.html
        - http://www-db.research.att.com/ode-announce.univ.html
        - A taxonomy for secure object-oriented databases
        - The BEAST Benchmark
    - Theory
        - The Object-Oriented Page
        - Cetus Links: Object-Orientation / OO DBMS
        - Object-Orientation FAQ -- Title Page
        - comp.databases.object mini-faq
        - The Evolution of Client/Server Architectures
        - Distributed Real-Time Systems (Bibliography)
        - Paradigms and Environments for the Development of Distributed Realtime Systems
        - Integrating Object Technology and the Web
        - Object-Oriented System Development: HTML Edition
        - Paradigms and Environments for the Development of Distributed Realtime Systems
        - OBJECT-ORIENTED DATABASE MANAGEMENT SYSTEMS - Introduction
        - Fine-Grained Sharing in a Page Server OODBMS
        - Ron Resnick's Distributed Objects and High Availability Page
        - Distributed Objects on the WWW: A Position Paper
    - Choosing an OODBMS
        - Objectivity White Papers: Choosing an ODBMS
        - Object Databases: Objectivity Published Papers
        - The Object Database Handbook : How to Select, Implement, and Use
        - Barry & Associates, Inc.

- Object-Oriented Databases
- Accessing a Database Server via WWW

APPENDIX B

Categories of time-based Media

The following lists provide several different methods of categorising time-based media resources. In building an archive, categories will be relevant for access and manipulation. In catalogue systems, these categories appear in the data sets of the Metadata. In relational databases, certain fields would contain the necessary category types to ensure the handling of the relations between them. In object-oriented databases, these categories might be realised as attributes, or even better yet, as inheritance, inclusion or association relationships between objects. The categorising of single entities of data, i.e. single independent pieces of information, is required for representing knowledge domains. Knowledge can be defined as pieces of information put in context to each other.

Even though a "Dublin Core based Taxonomy" is given as a last alternative, these categories should not be directly associated with the act of storing metadata. Categories are rather a way of storing information in a structured environment to be able to efficiently maintain and keep the context of its content involved. If possible, these categories should support searching and accessing mechanisms.

The Dublin Core metadata sets, or any other metadata sets, are a powerful way for accessing information by "search and retrieval" of metadata. Categories, or schemas, or implemented taxonomies, on the other hand, provide a powerful means of representing the real-life relationships between the resource objects (vital for the implementation process) and using these to browse through and close in towards specific areas in the research process of a future user.

Following are three alternative taxonomies of time-based media. The first alternative represents how the user, wanting to access and use an archive, might envision resource categories. Searching for a certain document or object, he/she will probably first want to define the discipline, and then the nature of the resource. The second alternative emphasises the type of the resource, and another with setting up a scale of complexity. Both of these concentrate on the resource itself. The last represents the Metadata approach, keeping close to the Dublin Core, but additionally specifying subtypes for in-depth representation of Metadata.

Discipline-based categories for TBM resources

- **music**
    - primary resource: representations of the work itself
        - recordings of the work
            - recording of a first performance of the work
            - recording of a performance with the composer
            - recordings of other performances
        - facsimiles of the work
            - facsimile of score written by composer
            - facsimile of first publication of the work
        - video as the work
        - complex-music-data data as the work(e.g. MAX)
        - scripts/texts as the music resource (e.g. action/text based scores)

        - graphics as the music resource (e.g. graphic scores)
        - printed score
    - secondary resources: resources about the work
        - recordings of
            - interviews
            - features
        - facsimiles of letters/notes containing information about the work
        - video as secondary information (composer, performance, etc)
        - image as secondary information (composer, performance, etc.)
        - reviews
        - analytical texts, diagrams and images
- **moving image**
    - primary resources
        - the resource film, if existent
        - all footage created during filming...if available
        - screenplay
    - secondary sources
        - reviews
        - descriptions
        - abstracts
        - information about the
            - producer
            - screenplay author
            - actors
            - music (see music)
            - author of the book, on which it might be based
    - related sources
        - book, on which the film is based
- **broadcast arts**
    - radio/tv/drama
    - reading
    - documentary
- **theatre**
    - primary sources
        - script

        - recording of the performance
    - secondary sources
        - reviews
        - descriptions
        - abstract
        - information about the
            - script author
            - actors
- **dance**
    - primary sources
        - choreography
        - notation
        - recording of the performance
    - secondary sources
        - reviews
        - descriptions
        - abstract
        - information about the
            - choreographer
            - dancers
            - script author

Resource type oriented categories

- **sounds**
    - sounds as the resource
    - sound as part of a resource
    - sound to underline secondary information of the resource
- **moving image**
    - moving image as the resource
    - moving image as part of the resource
    - moving image to underline secondary information about a resource
- **text**
    - text as the resource
    - text as part of the resource
    - text to underline secondary information about the resource

- **images**
    - images as the resource
    - images as part of the resource
    - images to underline secondary information about a resource
- **composite data** (data that is made up of a combination of the above given resource types)
    - theatre works
    - broadcasting features
    - film and soundtrack
    - "multi"-media works

Categories relative to complexity

- **simple resource elements**
    - midi
    - sound
        - wav
        - ra
        - etc
    - text
        - simple: non formatted
            - ascii
        - non simple: formatted
            - rtf
            - html
            - proprietary formats
        - non simple: other
            - SGML
    - video
        - mpeg
        - motion JPEG
        - ITU-601
    - image
        - pixel based
        - scalar based
- **resources of medium complexity (the resource is made up of a linked system of simple resource elements)**
    - multi-media applications as the resource

        - music applications made up of complex music structures (e.g. MAX scores)
- **resources of high complexity**
    - hyperlinked web applications as the resource
    - interactive multi-media applications as the resource

Metadata oriented approach

Even though a "Dublin Core based Taxonomy" is given as this last alternative, these categories should not be directly associated with the act of storing metadata. Categories are rather a way of storing information in a structured environment to be able to efficiently maintain and keep the context of its content involved. If possible, these categories should support searching and accessing mechanisms. Thus this approach chooses the Dublin Core in the first level as a means of categorising.

- **Resource**
    - subject
    - title
        - short title
        - full title
        - title of the work it belongs to (e.g. adagio of the Flute concerto e flat, or an essay title and the collection in which it appeared)
        - image of the cover page
    - author(s) or creator(s)
        - name(s)
            - first name
            - surname
        - type of creator
            - author
            - composer
            - artist
            - photographer
            - etc.
        - additional resources (information) about the creator(s)
            - text about authors' life
            - personal notes about his own work
            - portraits of the creator(s)
    - uncontrolled term
        - descriptions of the content of the resource
        - abstracts
    - publisher

        - name
            - first name
            - surname
        - company name
        - address
        - additional information about the publisher
            - text about publisher's life
            - personal notes about the work
            - portraits of the publisher/pub lishing house
            - communication between author and publisher
    - other contributors
        - name
        - type of contributor
            - editor
            - transcriber
            - arranger
            - sound-technician
        - additional information about the contributors
            - text about contributors' life
            - personal notes about the work
            - portraits
    - date (date of publication)
    - resource type
        - theatre
        - music
        - broadcast art
        - dance
        - video
        - film
        - music
        - multi-type piece of resource
    - format of the resource
        - digital resources
            - mime types
                - midi

            - wav
            - ...
            - jpeg
        - non-mime types
            - max
            - application
    - non-digital resources
        - book
        - painting
        - cassette
        - ...
- resource identifier
    - type of resource identifier
        - URL/URN
        - OID (Object identification. if existent)
        - ISBN
        - ISMN
        - archive identifier
- source
    - original work
- language
- relation
    - "has a" relationships (structural data)
        - is-part-of relationship
        - has-parts relationship
    - association
- coverage
    - place
    - time
- rights management
- terms/conditions
    - charging information
    - accessing information

Appendix C

How do users search?

It should be taken into account that search methods differ exceedingly, depending on who is searching, which resources is being searched for, what archive is being searched in and most of all, with which tools the search is supported. This page tries to discuss a variety of different search methods thinkable.

Search-and-Retrieve approach

The user knows some terms of the metadata elements and uses them to retrieve all corresponding resources that match these search terms. The simplest form would be searching for words in a text, the more complicated forms the defining of terms and subterms, (as for instance Dublin Core elements and its subtypes). Another example is the use of "filters", where all not matching data sets are filtered out, leaving matching hits.
All of these methods include the action of putting text in, and getting text back as the result, which might or might not be linked directly to the source being described by the text. It always involves the use of metadata.

Closing-in-on-data approach

A proper name has yet to be found for this approach, moving from the general to the specific. An example would be a user searching for a composer's works published under a different name. Starting to search into the direction of contemporary composers, reducing the search to a region, then to all that were known to use aliases, then to even reduce the result list further by specifying certain characteristics of the compositions. This example shows a case, where a search-and-retrieve method would have to be quite into depth to be able to get correct hits, or it would have to be given as general as possible, and have a huge hit number as a result. The circular approach, the closing-in on hit results, that have a high chance of being correct, is used unconsciously by everyone going into a normal library and looking for a certain date or element or fact in a book. Search methods in a resource environment should take it to account.

Browsing approach

Information includes its own different types of relations between resource objects. Data alone does not make sense. It is the relation of data towards other data which makes information out of these singular data sets. The way of linking sets of data to other data in the past and present library systems was to use some kind of metadata,, i.e. UK-Marc, US-Marc, etc. In many of the information systems today, browsing has become an important method of searching . Users browse through file-systems, users browse through hyper-text links, and developers use browsing facilities to access modules or objects or classes to program with. One technical prerequisite for browsing is implementing these different types of relationships. A normal file-manager system implements the "inclusion" relationship, a language browser for C++ or Smalltalk or Java implements the "inheritance" relationship, a VRML application may implement geographical relationships to be able to browse data in terms of movement. There might be different relationships that might be implemented in a browsing facility, such as:

- *Inclusion* - one object is included in another object
  (ex.: a file in a folder, a certain sound used in a composition a note in a bar)
- *Inheritance* - one object inherits the characteris tics of another object
  (ex.: all Bach works have a BWV Verzeichnis, so each single works inherits the attribute BWVerzeichnis-number of the Bach Works Object,
  or all service provider users have read rights, these might be inherited down towards the developers of collections, who also have write rights
  or as a third example all sounds stored in a high quality inherit the characteristic of being only served out on ATM in real-time)
- *Association* - one object is associated with another object
  (ex.: "Fingals Cave" the composition resource is associated with the geographical rock formation

of Staffa, or any other thinkable relationship between two resources)

- *Attributes* - an object contains certain attributes, or certain characteristics
  (ex.: all objects in the PADS archive have the attribute DublinCore, whereas the DublinCore object itself has 15 further attributes defining the elements of the DUblin Core
- *Others*

Demonstrate-and -Retrieve approach

Another seemingly strange but realistic approach, especially for the performing and visual arts, would be the "demonstrate-and-retrieve" method, demonstrating that, which the user is searching for by actively giving an input. For instance, a dance choreographer may want to search for a certain tempo and rhythm for his dancers, and may just want to tap them in and find music fitting to the rhythm, or an art historian may want to find all pictures painted with a certain colour , because he may want to analyze, for example, if it was true that most of the painters of north Jutland did paint with a certain 'bluishness' in their paintings. He would point to the colour looked for and try to retrieve it from a collection of paintings.