DATABASE REQUIREMENTS FOR DISTRIBUTED MULTIMEDIA SYSTEMS

Carlos M. TOBAR Toledo*
Ivan L. M. RICARTE**

ABSTRACT

Databases have been supporting many classes of applications with very distinct requirements. Multimedia is one class of application that has been receiving a fair amount of attention lately, with the increasing processing power of computational systems. Databases and multimedia come together when the multimedia application requests storage of or access to multimedia data. However, database-related requirements of multimedia, applications are yet to be completely understood and fulfilled.

This paper presents a set of requirements that must be provided by a database system in order to support the representation, storage, and manipulation of multimedia documents. These requirements provide guidelines to evaluate the adequacy of database systems for supporting multimedia applications. The main focus is on distributed systems, which reflects the trend towards the future of multimedia systems, whose needs could hardly be provided by stand-alone systems.

KEYWORDS: Multimedia Databases, Multimedia Information Systems, Distributed Systems, Multimedia Persistent Storage

1 - INTRODUCTION

Recent technological developments have allowed the construction of a continually growing number of multimedia systems. Such systems usually incorporate tools to capture and present digitized data either at stand-alone or networked systems. As a consequence, there is a growing interest for the provision of adequate mechanisms for the persistent storage, search, and retrieval of multimedia information. Most multimedia information systems still use isolated files to store multimedia data, since existing DBMSs do not fulfill the requirements placed by such applications. Ideally, Multimedia Database Management Systems (MMDBMS) would provided this functionality.

Although there are a large number of applications that could be classified as multimedia information systems, we are particularly interested on those that manipulate persistent objects containing multimedia data. Such objects are the multimedia documents, and they are the elements that are going to be stored and retrieved by MMDBMSs. The manipulation of such multimedia documents will also demand that such MMDBMSs should be distributed over computer networks, since most applications that are expected to benefit from the incorporation of multimedia resources, such as CSCW (Computer Supported Cooperative Work) and Office Automation, are implicitly distributed.

In this paper, we analyze how the requirements of distributed multimedia applications impact on design

(*) Institute of Informatics, PUCCAMP P. O. Box 317, 13020-904 Campinas (SP), Brazil tobar@zeus.puccamp.br
(**) Dept. of Computer Engineering and Industrial Automation, FEEC/UNICAMP P.O. Box 6101,13083-970 Campinas (SP), Brazil ricarteCdca.fee.unicamp.br

decisions related to a multimedia database system. Section 2 presents an overview of the general requirements present on multimedia applications and related work on supporting multimedia applications with database systems. Section 3 presents the structure and activities related to multimedia documents, and how such aspects impact on the multimedia database system. Section 4 addresses particularly the major operational requirements imposed by the exhibition of continuous media, the most demanding in terms of computational resources, and their implications to an MMDBMS. Section 5 presents a concept which introduces a major difference between the manipulation of conventional and multimedia data - quality of service (QoS), along with requirements that QoS imposes to an MMDBMS. Section 6 describes how metadata can be used to describe the nature of multimedia documents and to support the requirements imposed to an MMDBMS. Finally, Section 7 presents our conclusions.

2 - DATABASE SYSTEMS AND MULTIMEDIA

Some of the requirements imposed by the manipulation of multimedia data are the ability to handle many different types of media, to provide large amounts of storage space required by some media data types, and to provide large and variable storage and retrieval rates to transfer multimedia data. These requirements are detailed in Subsection 2.1. Subsection 2.2 presents an overview of related work, that shows that most existing database management systems do not satisfy such requirements.

2.1 - Requirements of Multimedia Data Management

When a multimedia document is viewed in the domain of an information system, it could be modeled and manipulated in two distinct ways [8]: as a real world object, named multimedia object, with properties and relationships, about which could be useful to obtain information; and as an indicator, recognized by the user, to real world objects, whose value, named multimedia content, could be used in the search for information about the pointed objects.

Specialized data, structures, rather than the BLOB type (binary large object) available in some DBMSs, are necessary in MMDBMSs to allow the manipulation of multimedia objects. An MMDBMS should allow storage of data with different types of media, and must also support definitions of new types. Metadata is required to describe instances from these types, thus allowing the use of semantic indices and processing of ad hoc queries.

Multimedia data pieces could require huge amounts of space in order to store data items such as a detailed color image and digitized stereo sound. An MMDBMS must provide adequate mechanisms to provide large data space and appropriate retrieval bandwidth for multimedia contents. In order to save storage space, mechanisms such as compression and sharing should be supported. Sharing [12] can occur in several levels, either by attending several requests at the same time, or by sharing multimedia components among several different multimedia documents.

Stereo audio and high quality video would require huge amounts of main memory in order to allow their processing as a whole. Since only some tens of frames or samples of data are needed to be exhibited per second, there is no need to store the whole piece in main memory, but retrieval must be done continuously. Actually, the flow is not really continuous and some codification techniques produce sequences with heterogeneous lengths, and the retrieval bandwidth may be variable.

An MMDBMS, just as a distributed DBMS [23], should provide persistence, sharing protection, concurrent access, reply performance, scalability, availability, mobility, information privacy, application support, and mainly data independence. The ACID properties essential to a DBMS, have in an MMDBMS a difference regarding the atomicity of requests, as addressed in Section 5.

2.2 - Related Work

Smith and Zdonik [24] realized one of the early studies comparing the use of relational and object-oriented databases in the context of hypermedia applications. They implemented the storage of application links using a relational system, and concluded that an object-oriented model would be more adequate for this kind of application. Woelk and Kim [26] present a model for a multimedia application, describing the implementation of a Multimedia Information Manager on top of the ORION DBMS. These pioneering papers have shown how limited classes of multimedia documents could be stored and managed using databases.

Other related work also favors object-oriented models for multimedia applications. Meghini, Rabitti, and

Thanos [19] describe conceptual modeling of documents in the context of a Multimedia Office Server with emphasis on retrieval operations. They also advocate the use of object-oriented data models as the better way to describe the structure of such documents. Ishikawa et al. [11] present an object-oriented knowledge base management system with support to static multimedia data, but not to temporal multimedia data, such as video and audio.

Grosky [9] provides an overview on Multimedia Information Systems. He proposes a generic architecture for such systems with three interrelated repositories of data, namely a standard database (with non-multimedia data), a MMDBMS (with uninterpreted multimedia objects), and a feature database (with information that could potentially help the retrieval of multimedia data). This approach addresses the fact that a single system can hardly provide efficient manipulation of all types of data. However, the proposed architecture does not address distribution aspects of such systems.

Other interesting aspect is user interaction with the multimedia document. Bulterman (3) analyzes multimedia document behavior during user interaction on a distributed multimedia application, and presents an infrastructure to support requests of real-time dynamic presentations of multimedia adaptable documents (that can be adapted to match available resources). However, the author does not address the problems related to data transfers. Sakauchi [22] addresses content-based vision and image retrieval, presenting a prototype using an object-oriented DBMS with a dictionary for video scene descriptions. Data retrieval is based on keywords describing an image (metadata).

Standards play an important role on the manipulation of multimedia data. Koegel et al. [15] developed a distributed multimedia information system using a multimedia standard, HyTime. This system constitutes one of the first efforts to validate the operational model of a multimedia standard using an object-oriented DBMS to store document descriptions. Its three-layered architecture presents information of hypermedia content and structure, object instances, and application objects and their attributes. Klas, Löhr, and Neuhold [14] present a tutorial on multimedia and its application requirements. Some multimedia standards are considered and integrated to a distributed and open extensible object-oriented DBMS system. They address some important features required by multimedia applications and not supported by object-oriented systems, such as time-dependent data support, user interaction mechanisms, and content-based query and retrieval techniques. Tobar and Ricarte [25] propose a distributed architecture for an MMDBMS, based on RM-ODP[1] and OMG CORBA[2] specifications, where data and metadata are stored separately. Questions were raised regarding the ideal allocation of services related to multimedia activities; some of these services could be supported by both a specialized component of the distributed platform and by an object database system. A framework to define allocation strategies is presented, and the role of multimedia-related standards, mainly MHEG[3], is emphasized.

3 - MULTIMEDIA MANIPULATION

There is a dichotomy inherent to the manipulation of multimedia objects through a MMDBMS. On one hand, the multimedia document is a real world object, as any other document on paper. On the other hand, the MMDBMS also states information about the document, which are abstractions of the real world object. Thus, a MMDBMS is both the place where artifacts are planned and designed (such as CAD objects) and also the place where these same artifacts are stored during and after their construction. A MMDBMS is a source of information to the exhibition of multimedia, documents and also is their repository.

3.1 - The Multimedia Document

For the user, a multimedia document is a single piece that can be read, heard, watched, or felt. Additionally, multimedia documents may allow user interaction with the exhibition vehicle, in order to enable control of choice for content, form, exhibition velocity, and quality. A multimedia document corresponds to a set of information pieces, each of which is represented through the use of different types of media, such as text, graphics, images, photo, video, animation, audio, sound, voice, music, or any combination of these.

Data components of a multimedia document could present several different interrelationships: temporal, spatial, accessibility (from what data piece in exhibition is possible to reach another data piece), selectivity (which data pieces must be considered for exhibition), and influence (which data piece, once manipulated, causes modifications in itself or in other data piece).

(1) Reference Model, Open Distributed Processing.
(2) Common Object Request Broker Architecture from the Object Management Group.
(3) Multimedia and Hypermedia information coding Expert Group.

The basic functionalities of an MMDBMS must correspond to the same basic functionalities of a conventional DBMS: registration and storage, search and query, modification and deletion, and retrieval. A MMDBMS should also support all the human activities related to multimedia. There are four major activities which are concerned with the production and use of multimedia documents, namely capture, authoring, perception, and management.

3.2 - Capture

Capture refers to the acquisition and codification of external data (to the computer) in a usually standard form. Once captured, the data piece is stored in specialized repositories, possibly integrated to an MMDBMS.

Usually a simple type of media data is captured using a specific codification and (optionally) compression technique, through hardware and software tools, creating pieces of multimedia content with some metadata that could be automatically generated. Even the simple capture of text could require specialized tools to capture meta-information such as structure, author, and date.

Continuous data need to be digitized before stored. The MMDBMS design should provide real-time support to store continuous data, preferably without quality loss. Furthermore, data should be adequately placed on disk in order to allow proper retrieval. Unlike DBMSs, the velocity of capture and retrieval imposes requirements on the performance of storage mechanisms of a MMDBMS.

3.3 - Authoring

Authoring refers to the development process of multimedia documents. Several types of software tools support and help planning, design, implementation, test, and maintenance of multimedia documents.

Multimedia documents are highly structured aggregates of simpler components, which can be see as multimedia documents themselves. Each multimedia document could be a simple media piece of data and metadata or a complex object. Even a simple media piece, such as a video, may be composed by several data pieces (images and sounds). Data pieces containing time-based media may contain derived data, defined as functions of real frames or samples, and generated only when needed, saving storage space and transmission time.

A MMDBMS should provide a data model for the representation of both derived and non-derived data and their relationships [7]. Composition is the way for the specification of relationships between data pieces and components of a multimedia document. Several types of relationships should be supported by the MMDBMS.

The production of a multimedia document probably involves several compositions and derivations. The authoring process, after the capture process, is supported by a database system [7]. It is usual to have incomplete multimedia documents stored, along with several versions, complete or incomplete.

An individual multimedia document could be exhibited in several different forms, named interpretations or views, due to several reasons: security, availability of resources, user selectivity, or operational problems that could happen during an exhibition. The MMDBMS should provide mechanisms for the storage and retrieval of interpretations during the development process. An interpretation should identify the components that actually will be exhibited and their relationships, along with information such as codification formats, duration, and location. Some interpretations should be adaptable to the dynamics of a distributed environment.

3.4 - Perception

Perception refers to the use of the existing information in a multimedia document, through its exhibition, using hardware devices for decoding multimedia content. Support of interaction is optional, involving specific hardware devices.

In addition to the normal presentation operation of continuous media (play), backward, fast-forward, fast-backward, random access and pause are desirable. All these selective user choices require variable data rates by the retrieval mechanism of the MMDBMS and by the communication transport mechanism.

Some exhibitions could involve data geographically distributed, with different types, which should be adequately retrieved in a synchronized way in order to allow proper processing. Both exhibition preprocessing and presentation require metadata in order to be realized. Those metadata should be stored on and retrieved from the MMDBMS.

3.5 - Management

Management refers to the planning, monitoring, and control of the computational environment, including all hardware and software resources, in order to allow the adequate execution of the support mechanisms for multimedia document manipulation. Aspects related to management are configuration, fault handling, security, service billing, and performance [13]. Quality of service (Section 5) should also be related to the perception activity.

Other management aspects are involved with the capture and authoring activities, very similar to management of software development. Metadata and statistical data, provided by the MMDBMS, are necessary for the management activity.

4 - MULTIMEDIA OPERATIONAL SUPPORT

Continuous (time dependent) data is the most resource demanding type of media. Considering this worst case for the operationalization to exhibit a multimedia document it is necessary to:

(i) retrieve the multimedia content along with information about its characteristics, both of them could be spread through the network;

(ii) transmit data chunks to the exhibition node, forming flows of data (there could be more than one, related to different media components);

(iii) decompress and process (to originate derived data) the data chunks;

(iv) synchronize and temporize the data flows;

(v) decode the received data originating the planned stimuli; and

(vi) process user interactions, such as selective choices and data modification requests.

All the six steps occur continuously, in parallel, and must be synchronized for a given multimedia document. In a typical environment there may be several multimedia exhibitions happening at the same time, possibly involving different multimedia documents. A new request should not be served if it would, by any means, impact requests already being served.

When a request is made, a scheduler should retrieve operational information in the MMDBMS in order to verify if there is the minimum amount of resources necessary to attend the request. Operational requirements could be adequately treated through mechanisms which allow high data bandwidths, a binding process based on media types, and effective search requests based on multimedia content, which are discussed below.

4.1 Hardware and Software Bandwidth Solutions

There is no commercial available technology to support the bandwidth requirements for multimedia exhibition in a multi-user, multi-task, distributed environment. Adequate support should consider high resolution quality request, variable data rates, and eventual peaks of bandwidth demand such as one caused by a request for a fast-forward presentation.

User requests are served in a continuous way through cycles of resource scheduling. At the beginning of a cycle, one request of each resource queue is scheduled to be processed in an integrated way, forming a hierarchical scheduling process composed by the bandwidth of network, memory, disk controllers and devices [1]. In addition there should also be a CPU scheduling for preparation and exhibition steps.

There are many techniques and mechanisms to solve the bandwidth problem which have been proposed in different contexts. A MMDBMS should support a combination of some of these techniques in order to support the demand of multimedia applications. Among these techniques there are: disk stripping of a multimedia component through several disk devices, allowing the bandwidth aggregation of each device [6]; optimized disk layout, including surface and cylinder allocation space [17]; minimization of disk mechanical movements; buffering [20]; pre-fetching [4]; hard real-time scheduling [20]; data distribution [6]; tertiary storage [10]; request batching [12]; device sharing [6]; data replication and resource dedication [6]; clustering [1]; resource management [1]; admission control [1]; and frame omission [1].

In addition, multimedia content could be stored in non-conventional ways in order to obtain better performance. Some techniques for specialized content storage are interleaving different multimedia contents to facilitate synchronization, padding (introducing gaps along with data to achieve a regular transfer data rate) and storing items out of order (to optimize compression schemes). A combination of these storage approaches could be used in a MMDBMS.

4.2 - Multimedia Types and Descriptors

A multimedia document, while a computational element that is transmitted and processed, presents type and value (content), such as discussed for the layout model (Section 6). The minimum information that is needed to know about a multimedia document, for its proper perception, includes its type and codification attributes. This information is named *media descriptor [7]*.

A media type is an specification of the attributes that could be found in media descriptors, along with their possible values. For time dependent media, the type also allows the specification of descriptor formats for different content elements.

Media descriptors should have information to help resource allocation for the exhibition. They could include average data rate, measures of data rate variance, and parameters for decompression.

The multimedia designer could use available types to describe rigid or flexible requirements for the exhibition devices. If the choice is to use flexible types, the exhibition application can obtain compatible conversions. The type describes the designer intentions regarding possible interpretations.

A type system should allow the specification of (abstract) structure and semantics of the transmission formats, in addition of being extensible to accommodate new formats (5).

4.3 - Location Indices

Existing storage systems do not provide satisfactory forms for information inference about data content, implying usually in the manipulation of whole documents. In a MMDBMS it is not realistic to expect a complete transcription of every content in form of metadata.

Time and spatial indices could solve this problem. An index could be generated automatically, if there is a pattern matching mechanism and a collection of patterns; or it could be planned manually whenever required. Examples of indices are location of key words in a textual image, location of words in an audio or speech, or objects in an image. Sometimes it is not possible to process a match to a pattern because of environment disturbances, such as noise or intonation in a speech. Thus it is interesting to generate a confidence score along with the result of a search.

5 - QUALITY OF SERVICE

Perfomance granting provided to user requests is generally referenced as quality of service (QoS). A service request only can be accepted if there is granted resource availability to its attendance, and there is no violation on the assured QoS of on-going requests.

QoS refers to the capacity that can be found in storage and communication systems in order to provide better costs and performance [18]. A QoS policy can only be implemented when based on a media type system or similar mechanism. Thus metadata plays a major role during operational control.

Flow channels are created between the flow source and destination interfaces, when they are compatible. The computational activity related to the creation of this channel is named binding [5]. A binding trial should fail whenever there are not available resources. If a combination of devices and multimedia, descriptors cannot be found, it can be concluded that the required QoS cannot be satisfied.

Admission control and scheduling policies are good mechanisms to grant QoS to user requests. Unlike DBMSs, a MMDBMS cannot afford performance degradation, thus it is preferable to postpone the attendance of a user request than serve it with poor quality.

QoS specification for an individual user request may be flexible, requiring specific resources but accepting eventual degradation without disturbing other on-going requests. This degradation is a multimedia-specific way of treating errors, since from the MMDBMS point of view there is no need for conventional transaction mechanisms. If there is any chance of anticipating QoS problems, the request service is postponed, but if it is served, it will be from the beginning to end, without rollbacks. Any unexpected problem should be treated by the exhibition server according to the multimedia type constraints.

6 - MULTIMEDIA METADATA

MMDBMSs should provide support for four distinct types of information models, which together could represent the dichotomy related to stored multimedia documents. A structural model allows modeling of multimedia objects, i.e., it allows the specification of the components of a multimedia document along with their complex relationships. An application model makes it possible to model the characteristics of the real world objects described through the multimedia content. A layout model allows the definition of interpretations for an multimedia document, supporting the specification of the external views perceived in an exhibition. Finally, a conceptual model allows the specification of the representation description used for data and metadata of a multimedia document[4].

Every generated information from these four models constitutes metadata. The structural and layout models correspond to the difference between internal (logic) and external representations. Layout data is partially derived, being generated during an exhibition retrieval. Most of the layout information is also structural information. QoS is directly related to the layout model.

The application model allows the representation of the relevant features related to a multimedia document which could be used by the user to infer information about the multimedia content, and by the MMDBMS to retrieve data and metadata. The conceptual model allows the specification of the descriptions used in the other three models. Through this model it is possible to capture information about metadata, i.e., meta-metadata. This second level of meta-information is necessary in order to process types, descriptors codification and compression patterns.

A possible metadata classification [2] comprises the following seven types of metadata. Document composition is related to the structural model, describing relationships among components of a multimedia document. Content location is related to the structural model. Multimedia content could be stored in specialized repositories, controlled by the MMDBMS, and in turn could be distributed, replicated, or stripped. Content description is related to the application model, describing for example experience level of the user for content manipulation. Document history is related to the application model, describing for example the list of persons or institutions related to the document, event location, price, or description language. Content classification is related to the application model, describing for example the status of multimedia components in a publishing environment (approved, in development), author name, composer name, or visit occurrences to the multimedia document. Statistical metadata is related to the application model, describing information, about multimedia, document collections, which is not necessarily stored data (could be derived on demand). Media type representation is related to the conceptual model and includes format, codification technique, compression method, resolution, and colors.

Metadata related to the layout model can be classified as exhibition description (comprehends necessary resources, quality information, failure resolution, and influence) or *exhibition content* (comprehends the data that really will be exhibited). The exhibition description corresponds to the multimedia type to be used for the transmission and exhibition processes.

A further classification for the conceptual model could include media type codification and compression and metadata meta-descriptors, which comprehends type descriptions and other similar information. Another class related to the application model is *document security* for the specification of access constraints.

7 - CONCLUSIONS

This paper presents a set of requirements for the proper design and development of an MMDBMS. This new type of data management systems cannot be completely implemented using conventional DBMS technology.

In a MMDBMS there is data and metadata sharing the same storage space, besides derived and interpreted data, context, and semantics. The space management should be oriented towards performance factors. Performance, and QoS by extension, is one of the more important issues of multimedia applications, multimedia repositories, and MMDBMS.

Multimedia data, without the existence of metadata, are semantically incomprehensible. Thus they require understanding, which usually is an application responsibility. A MMDBMS cannot be just a data repository with huge space capacity and high and variable retrieval data rates, it also need to provide semantics to the stored data. Metadata is defined as derived properties of a multimedia document which could be useful to access and retrieve multimedia content and are also necessary for its manipulation. Metadata is one of the major differentials between a DBMS and a MMDBMS.

Four models of information are necessary for the complete specification of multimedia documents. This classification extends the proposal by Meghini and others [19] with the inclusion of an application model. The metadata classification also extends previous work [2] by exploring and complementing the types of metadata in order to cover the four information models.

MMDBMSs ought to be integrated in an open way with the other multimedia services, mainly communication and exhibition. In order to enable this integration,

(4) The term conceptual in this sense was used by Meghini, Rabitti and Thamos [19]. Although traditionally the concept behind the structural model is named conceptual model in the database literature, their original terminology is adopted here.

multimedia type and service interface standards should be developed. Other support services and facilities available in an open and distributed environment, such as trading, brokering, and billing could use and update metadata in the MMDBMS.

An MMDBMS should provide a couple of specialized functionalities not found in conventional DBMS in order to allow proper multimedia document manipulation. In addition to supporting the requirements presented in this paper, a MMDBMS should be able to handle every available multimedia codification standard in the same way that it support different media types. Supporting standards allows the implementation of open systems [14] and is a compromise with technology rather than with vendors. Kwaaitaal, Hoogeveen and Van Der Weide [16] discuss the standardization of MMDBMS, presenting a reference model for the impact of standardization. They point out the lack of a standard multimedia database language for definition and manipulation of multimedia data.

These requirements are being used to govern the development of a distributed rnultimedia database system in the context of an open distributed multimedia plataform [21].

REFERENCES

[1] A. BILIRIS, B. K. Hillyer, and E. Panagos.The Calico multimedia storage system. AT & T Bell Laboratories, October 1994. 24 pages.

[2] K. BÖHRN and C. T. Rakow. Metadata for multimedia documents. *ACM SIGMOD Record*, 23(4): 21-26, December 1994.

[3] DICK C. A. Bulterman. Specification and support of adaptable networked multimedia. Published in Multimedia Systems,1993. ftp: //ftp.cwi.nl.

[4] J. K. Dey, C.-S. Shih, and M. Kumar. Storage subsystem design in a large multirnedia server for high-speed network environments. In IS&T/SSPIE *Symposium on Electronic Imaging Science and Technology*, February 1994. 12 pages.

[5] F. Eliassen and J. R. Nicol. Supporting interoperation of continuous media objects. *Theory and Practice of Object Systems*, 1995.

[6] S. Ghandeharizadeh and L. Ramos. Continuous retrieval of multimedia data using parallelism. *IEEE Transactions on Knowledge and Data Engineering*, 5(4):458-669, August 199:3.

[7] S. GIBBS, C. Breiteneder, and D. Tsichritzis. Data modeling of time-based media. In *1994 Proceedings* of the ACM SIGMOD Internantional Conference on the Management of Data, pages 91-102, June 1994.

[8] W. I. GROSKY, F. Fotouhi, and I. K. Sethi. Using metadata for intelligent browsing of structured media objects. ACM SIGMOD Record, 23(4):49-56, December 1994.

[9] William I. Grosky. Multimedia information systems. IEEE Multimedia 1(1):12-24, Spring 1994.

[10] W. B. HARDING, C. M. Clark, C. L. Gallo, and H. Tang. Object storage hierarchy management. *IBM Systems Journal*, 29(3):384-396, 1990.

[11] H. ISHIKAWA, F. Suzuki, F. Kozakura, A. Makinouchi, M. Miyagishima, Y. Izumida, M. Aoshima, and Y. Yamane. The model, language, and implementation of an object-oriented multimedia, knowledge base management system. *ACM Transactions on Database Systems*,18(1):1-50, March 1993.

[12] M. KAMATH, K. Ramamrithan, and D. Towsley. Continuous media sharing in multimedia database systems. In Tok Wang Ling and Yoshifumi Masunaga, editors, *Proceedings of the 4th International Conference on Database Systems for Advanced Applications* (DSAFAA'95), April 1995. 8 pages.

[13] B. KINANE. Distributing public network management systems using CORBA. In *Open Distributed Processing: Experiences with Distributed Environments, Proceedings of the IFIP International Conference, on Open Distributed Processing, ICODP'95*, pages 117-130, February 1995.

[14] WOLFGANG Klas, Michael Löhr, and Erich J. Neuhold. Multimedia, database systems. *In Proceedings of the Workshop on New Database Research Challenges*, pages 93-125, Rio de Janeiro, Brazil, September 1994. PUC-RJ/VLDB.

[15] J. KOEGEL, L. Rutledge, J. Rutledge, and C. Keskin. HyOctane: A HyTime engine for an MMIS. In *Proceedings ACM Multimedia*, August 1993.

[16] I. KWAAITAAL, M. Hoogeveen, and T. V. Weide. Reference model for the impact of standardization on multimedia, database management systems. *Computer Standards & Interfaces*, 16(1):45-54, April 1994.

[17] P. LOUGHER and D. Shepherd. The design of a storage server for continuors media. *The Computer Journal*, 36(1):32-42, 1993.

[18] S. LOUIS and D. Teaff. Class of service in high performance storage system. In *Open Distributed Processing: Experiences with, Distributed Environments, Proceedings of the IFIP International Conference on Open Distributed Processing, ICODP'95*, pages 323-334, February 1995.

(19) CARLO MEGHINI, Fausto Rabitti, and Costantino Thanos. Conceptual modeling of multimedia documents. *IEEE Computer*, 24(10):23-30, October 1991.

[20] A. L. NARASIMHA and J. C. Wylie. I/O issues in multimedia system. *IEEE Computer*, pages 69-74, March 1994.

[21] IVAN L. M. Ricarte and Carlos M. Tobar. Towards an architecture for distributed multimedia, databases. *In International Conference on Intelligent Information Systems*, Washington, DC, June 1996.

[22] M. SAKAUCHI. Database vision and image retrieval. *IEEE Multimedia*, 1(1): 79-81. Spring 1994.

[23] D. E. SHACKELFORD, J. B. Smith, and F. D. Smith. The architecture and implementation of a distributed hypermedia storage system. Technical Report TR93-013, University of North Carolina at Chapel Hill, 1993 16 pages.

(24) KAREN E. SMITH and Stanley B. Zdonik. Intermedia: a case study of the diferences between relational and object-oriented database systems. In *Proceedings of the International Conference on Object-Oriented Programming Systems and Languages*, pages 452-465. ACM, October 1987.

(25) C. TOBAR and I. Ricarte. Multiware database, a distributed object database system for multimedia support. In *Open Distributed Processing: Experiences with Distributed Environments, Proceedings of the IFIP International Conference on Open Distributed Processing, ICODP'95*, pages 439-450, February 1995.

(26) DARRELL WOELK and Won Kim. Multimedia information management in an object-oriented database system. In *Proceedings of the International Conference on Very Large Data Bases*, pages 319-329, Brighton, England, September 1987.