Multimedia Database Systems
-
The Notions and the Issues *

Thomas C. Rakow[1], Erich J. Neuhold[1,2], and Michael Löhr[2]

[1] GMD - Integrated Publication and Information Systems Institute (IPSI)
[2] Technical University of Darmstadt, Department of Computer Science
Address: Dolivostr. 15, D-64293 Darmstadt, Germany
E-Mail: {rakow, neuhold, loehr}@darmstadt.gmd.de

Abstract. In this article, we give an overview of the usage of database systems in the emerging field of multimedia computing systems. We motivate that management of multimedia information requires support of temporal relationships, interactive operations, high data volume, and transport of multimedia data. The solutions achieved so far vary to which extend multimedia applications are supported. We suggest to use the notions of hybrid, structural, behavioral, and distributed multimedia systems to classify them. We emphasize the current issues in the area of multimedia database research. Multimedia data modeling, content-based retrieval, continuous storage management, and a distributed architecture are the constituents of multimedia database management systems. The developments achieved so far with the AMOS prototype currently under development at our institutes serve as an illustrating example of current efforts. Specifically, the design of a datatype for audio information, the modeling of meta information and of interactive multimedia presentations, the development of an object manager for continuous objects, and the integration of an information retrieval system in our system are described.

1 Introduction

Most multimedia applications involve a diversity of conventional data types like numbers, text, and tables combined with media data like images (bitmaps), graphics, audio, video and animations. Early research results in the area of multimedia systems have shown that object-oriented programming is very promising to provide for the multitude of datatypes and their manipulation features. In multimedia system development a database management system can provide several useful services: transparency from physical aspects of storage, associative access through indexing, data consistency through defined access methods, query

* Arbeitspapiere der GMD, St. Augustin, February 1995. *To be published in:* Tagungsband GI-Fachtagung Datenbanksysteme in Büro, Technik und Wissenschaft (BTW), Dresden März 1995. Springer Informatik Aktuell, Berlin 1995.

facility with descriptive access, multi-user access through concurrency control and reliability through recovery mechanisms. In order to preserve these advantages, it is necessary to integrate multimedia data into database management systems. However, besides the concepts of object-orientation several other concepts need to be employed by a multimedia database management system.

Differently to ordinary data the presentation of multimedia data is not canonical. The types of conventional data usually comprise the types known from programming languages (e.g. character, integer, real or records of these types) and their representation is inherent to a computer system. Multimedia data are not directly supported by programming languages and their presentation depends on special devices as well as additional information such as image format, compression techniques and layout description. The benefits of database systems are especially found in areas were groups of information producers cooperatively create complexly structured multimedia information. This information must be stored and manipulated/updated over long periods of time and is accessed by a multitude of information consumers looking for "individualized" information that satisfies the consumer's information needs of the moment.

A multimedia database management system (MM-DBMS) should have the capability of storing, managing and retrieving information on individual media, managing interrelationships between the information represented by different media, and should be able to exploit these media for presentation purposes. Concepts for modeling time-dependency and synchronized presentation of multimedia data must be added and integrated into the data description and the query language. Furthermore, presentations and control of presentations at the user's workstation requires a client/server architecture, buffering concepts, and networks that support continuous or isochronous transport protocols. The necessary access to existing multimedia data stores (e.g. CD-ROM systems) requires flexible integration techniques.

One of the key obstacles for many multimedia applications is the vast amount of data involved. The use of digital images often is not viable due to high storage or transmission costs, even if image capture and presentation devices are affordable. Modern image compression technology can compress typical images from 1/10 to 1/50 of their uncompressed size without visibly affecting image quality. But the storage requirements for one object still exceed the size of an average object handled by conventional database systems. For example, objects may not fit into main memory in their entirety.

Traditional database applications use data of fixed size, but the size of multimedia data can vary dynamically. A frame in a video can be regarded as a single object and treated accordingly in the database, but this way continuous presentation is not feasible. So far all unformatted data (mainly text and images) has been handled in database systems through long fields or BLOB's (binary large object), but they usually support only a few generic operations such as reading or writing parts of long fields. Moreover, they are excluded from queries. It is necessary to impose some structure to accommodate for example a frame structure in a video sequence.

Within audio, video, and voice the information itself is expressed as a function of time. Synchronization assures a temporal order of events. Examples of synchronization of multimedia data are the playback of a movie with its soundtrack and the alignment of two stereo channels. Conventional databases provide neither mechanisms for expressing these synchronization conditions nor for controlling them.

In contrast to conventional data the production, manipulation, and presentation of multimedia data is performed with special devices or tools. To provide the flexibility needed, generic interfaces must be provided to integrate new devices. The access to these devices by concurrent capturing and presentation processes must be controlled and the interaction with them must be handled (i.e. interrupting the presentation of a continuous multimedia object). The integration of these devices is very appealing from another point of view. They enhance the possibility for user interaction with the database, for example interaction-supported queries (e.g. based on a pointing device) become possible. This can be achieved only partially in case these devices are directly accessed by the application.

1.1 Related Work

Research in the field of storing time-dependent data addressed new solutions for buffering, layout of files, and strategies for accepting requests in parallel of ongoing services. The proposed solutions can be classified into deterministic and statistical approaches regarding the performance guarantees. Deterministic approaches guarantee reliability of services, statistical approaches only provide service reliability with some probability. In the statistical case, unpredictable delays may occur because of resource contention. Another inherent problem is the acuracy of the statistical model. Deterministic approaches usually rely on worst-case scenarious and, hence, involve the risk of underutilized systems, because they often do not take into account variations in e.g., the compression rate of audio and video. For examples the reader is referred to [A+92, GC92, LS93] on continuous media file and storage systems, [YSB+89, WYY91] on efficient storage of audio data on optical discs, [RVR92, RV93] on support for on-demand multimedia services and techniques for interleaving multiple read requests.

Multimedia database systems are a relatively new field due to the fact that the necessary hardware became available only recently and is still developing [AK92, Gro94, MW94]. The first approaches were database systems for specialized data such as spatial databases [OM88], [SR86] and pictorial databases [TY84]. But in some cases a DBMS was only used as a somewhat complex file system. Spatial databases are attractive because the semantics of the objects and operations are clearly defined and their properties can be derived from geometry.

One of the first efforts in managing multimedia data was MINOS, the multimedia object presentation manager developed at the University of Crete [CHT86]. Another early project was the Multimedia Information Manager of the ORION object-oriented DBMS, developed at MCC [WKL87, WK87] and now available as the product ITASCA. The integration of the new datatypes is accomplished through a set of definitions of class hierarchies and a message-passing protocol not only for the multimedia capture, storage, and presentation devices, but also for the captured and stored multimedia objects. This way a high degree of flexibility is achieved since new storage or presentation devices are included easily by providing the corresponding types as subtypes of the existing types. This approach is very promising, but it remains more or less a collection of classes. Specific database issues such as query processing, user interaction and architectural implications are not considered. Furthermore, it can be questioned whether the modeling of devices down to the level of methods such as `get-next-block` lead to efficient realizations.

Sometimes, multimedia systems are designed as *federated* systems because heterogeneous information sources shall be integrated [B+90, Mas91] or conventional and multimedia information are stored in different systems [KMMW94]. The latter system called MOSS aims at the integration of a multimedia set-oriented data server with a relational DBMS. Cooperation of federated systems is a problem not only specific to multimedia systems. Hence, specific requirements for multimedia systems cannot be drawn from these systems. In [Loc88], at least different systems for discrete and time-dependent data were assumed.

At GMD-IPSI, research in the area of MM-DBMS takes place in the AMOS (Active Media Object Stores) department. The objective is the development of the concepts needed for a "true" MM-DBMS and their integration within the existing object-oriented database management system VODAK. Main topics of research are modeling of multimedia information [AK94, BA94, BR94, KNS90] and support for time-dependent data [RM93, RLM+93]. Prototypes were developed and used in applications in telecommunication environments [R+94, TR94]. The AMOS developments serve as an illustrating example of the features and the architecture of a MM-DBMS in this article.

1.2 Overview

In *section 2* of this paper we give a detailed motivation for specific requirements of multimedia data. *Section 3* describes current solutions and introduces the notions of multimedia systems. We emphasize multimedia data modeling, content-based retrieval, continuous storage management, and the architecture of MM-DBMS's. In *section 4*, the developments achieved so far for the AMOS prototype are observed. In *section 5*, we show how our system can be used within a sample application. This application supports the engineering process such that dependencies between different multimedia specification and report documents can be managed automatically and global consistency is ensured. *Section 6* concludes the paper with an outlook.

2 Multimedia Information

The characterizing property of multimedia information in general is the incorporation of continuous media like video, audio, animation together with conventional types of data. This section gives an introduction to multimedia systems as far as it is needed in following sections. Introductions to the basics of multimedia systems and their technical demands in general are given in [EF94, Fur94, Gro94, Ste93].

2.1 Datatypes and Formats

The notion *multimedia data* covers alphanumeric data as known from conventional computer systems, new types of data like *audio*, *video*, *graphics*, *images* (*pictures*), *speech*, *music*, *animation*, and any arbitrary composition of these data types. The new types of data add new dimensions to the properties of conventional data types and their handling by a computer system. First, data types like *audio* and *video*, often called *continuous data*, are *time-dependent*. Second, these new data types reflect an increased complexity and are no longer just symbolic data, i.e., they usually have associated compression techniques and need complex operations for their interpretation and manipulation. In the following we characterize some of the media data types in more detail in order to derive some important requirements for multimedia database management systems.

Image: In the past images were stored and distributed as drawings, paintings, photographs or prints. A multimedia system has to provide the functionality to import and manipulate digital equivalents of these media. Basic manipulation operations are clipping, scaling, chromatic correction and the composition of several picture sources. To handle image data a database management system has to manage large amounts of simply structured data efficiently. The internal representation of the images should be hidden to end users and applications by an adequate abstraction.

Graphics: The notion of computer graphics includes all concepts that allow to generate drawings and other images based on formal descriptions, programs or data structures. Typical elements of computer graphics are lines, regions and text elements. Several standards [ISO84] have been established and serve as the basis for industrial and scientific applications. The proper integration of graphics with other media types and existing systems as well as the ability to handle the complex-structured data representing graphical objects are the central requirements for a multimedia database management system in this context.

Text: For a multimedia system it is often not enough to process and represent text as simple strings of characters. A useful representation of textual information should include structural information like title, authors, authors' affiliation, abstract, sections, subsections, and paragraphs. An example of a standard which allows to express the logical structure of documents is SGML [ISO86]. In addition to the representation of the logical structure of text a comprehensive representation of textual information has to represent layout information as well. The complexity of all these structures, which arises from concepts like nesting and repetition, requires powerful modeling capabilities from an underlying database management system [BAH94].

Audio: A common property of all the media previously mentioned is their independence from time. In contrast to these time-independent data an implementation of audio has to consider its time-continuous characteristics. Audio data has only a meaningful interpretation with respect to a constantly progressing time scale. Manipulation operations like cut, copy, and paste can still be handled statically, but playback and recording operations will always be associated to a time scale. Time-dependency, the necessity to express temporal relationships, and the support for compression techniques need to be reflected by an implementation of an audio datatype and its associated operations in a multimedia database management system. This allows the database system to provide a basic understanding of the semantics of audio data.

Speech: The media type speech often is not recognized prominently in the context of multimedia systems. At a first glance it seems not necessary to handle speech as something different than audio, because speech is usually stored as audio data, but recent improvements in speech recognition allow to search for characteristic keywords [RJ93] and to identify specific speakers [WB92]. With advanced progress in the field of spoken natural language processing it will become more important with respect to the interaction features of multimedia systems. For example, speech data can serve as input for the retrieval of stored audio and speech data, or speech data can be generated as a result of queries. Advanced retrieval operations like a best-match word retrieval operator for speech documents [SG94] may be defined on audio data. These algorithms need additional abstractions which play a role that is comparable to traditional indexing techniques in information retrieval. Therefore, even if speech shares the characteristics of audio it shows some unique properties of spoken natural language in addition.

Video: Video combines the properties of the media types audio and picture. Like audio data, video data is time-dependent. The manipulation operations like cut, copy, paste, playback, and recording are similar to those defined for audio data. The atomic constituents of video data are video frames which are closely related to picture data. Advanced retrieval operators may be defined on the content of a video, e.g., retrieving particular portions of a video which start with specific scene cuts which are close to a given picture. These properties of video data have a significant impact on the implementation of video data types and the handling of the data by a database management system.

Generated Media: Examples for this category are computer-generated animation and music. If these media are computed in advance and stored as video or audio data there is no difference to other video or audio material. If they are generated in real time during presentation, both can be seen as a special kind of continuous media types. The approach to generate animations and music on the fly during presentation clearly increases the possibilities of interaction. Examples are changes in the visual angle on the scene presented or the manipulation of simulation parameters, if an animation sequence represents a simulation result. Presentation speed may be changed with less problems than it is for audio or video. The generation process can easily produce sounds of arbitrary length or generate additional pictures to present an animation in slow motion.

2.2 Temporal Relationships

Conventional documents are statically structured compositions of, e.g., text, graphics and pictures. Multimedia documents become dynamic as soon as they incorporate time-dependent media. The dynamics is based on the temporal relationships between the continuous media components that are part of a presentation. Examples for such relationships are playing back a video and an audio *simultaneously*, playing back two videos *in succession* or *after a predefined temporal interval*. While object-oriented database systems provide useful concepts to handle complex static structures, they are lacking the ability to handle the dynamics inherent to temporal relationships. The modeling and presentation of multimedia information, however, strongly depends on the ability to reflect temporal relationships. Static information is usually presented as a whole piece of data or, triggered by subsequent user interactions, piece after piece. The operations on time-independent data do not consume time and their execution can be modeled as a sequence of steps. In contrast to this, the presentation of dynamic media is expected to last for some time. User interaction at any time can change the presentation.

In order to cope with temporal relationships a multimedia database system must provide some notion for *parallelism*, e.g., for executing a playback operation on videos and audios simultaneously or to control user interaction in parallel to media presentation. The system should also provide for the representation of temporal relationships and their composition (see also section 4.4).

Synchronization: As soon as several time-dependent media are to be presented, it is no longer sufficient to merely express the temporal relationships between the continuous media and to lean on the concept of parallelism. Temporal coordination of several media and therefore concepts for *synchronization* become necessary. As the presentation of information to the end user belongs to the tasks of a database management system (at least in some default mode which is already needed if queries against multimedia data are allowed), a multimedia database system must support synchronous delivery of media data all over the way from the storage system to the user interface. Hence, it is necessary to model and store temporal relationships between the media data in addition to the data itself. Based on this information a synchronization mechanism is needed to handle various related continuous data streams during presentation according to temporal relationships defined between the streams.

Processing: A mechanism for the correct processing of temporal relationships should provide certain "concessions" with respect to the actual specification of temporal relationships. To describe the requirements of a multimedia application on one side and the performance of multimedia system components on the other side the notion of *quality of service (QoS)* has been introduced [LG90]. A complete QoS definition consists of the following parameters:

- *Average Delay* describes the time between the triggering event (e.g. user interaction) and the observable reaction of the system by means of executing an operation. For example, the average delay between submitting a query to a database system and receiving the (multimedia) result is a critical parameter for user acceptance.
- *Speed Ratio* is defined as the ratio between the originally intended and the actually achieved presentation rate. This parameter relates the actual presentation speed to real time and therefore allows the specification of increased or decreased playback speed.
- *Utilization* describes the ratio between the amount of media data used for the actual presentation rate and the total amount of data available for this presentation. For example, using only 8bit out of 16bit audio information corresponds to a utilization of 1/2.
- *Jitter* is a measure for the temporal deviation of two simultaneous presentations at a certain point in time.
- *Skew* is a measure for the accumulated temporal deviation of two simultaneous presentations during a certain interval of time [SE93].
- *Reliability* describes the average frequency of errors during a given time interval of media presentation or recording. Reliability may be measured at multiple levels such as bits, packets, or whole frames.

The parameter settings vary depending on the concrete applications. The parameters themselves are subject to complex interactions depending on the QoS requirements of a multimedia application.

2.3 User Interaction

User interaction becomes much more complex if multimedia data is involved in a presentation for a user or dialog with a user. The state-of-the-art concepts like buttons, text entry, scrollable areas for constructing user interfaces do not support the interaction with continuous media. New devices like cameras, microphones may be taken into account in addition to keyboard, mouse, and external equipment like VCRs for input recognition (speech and gestures). Presentations may be directed to various devices like windows, monitors, loudspeakers. To handle multimedia data additional presentation-related services and concepts are needed [RLM+93]:

- *Simultaneous control of different devices*, i.e., support for temporal composition and synchronization so that timing constraints between different presentation steps can be handled. The control of these devices may be based on high level abstractions in order to decouple the database system and application from device-specific and low-level synchronization issue.
- *Efficient handling of user interrupts* is crucial for multimedia applications as the interaction with the user is based on a visual (video) and acoustic perception due to the usage of time-dependent data which is much more sensitive to timing than other interaction styles.
- *Standardized interaction paradigms* for multimedia-related interaction like controlling a VCR, continuous media streams and image presentation should provide for a universal interaction paradigm with a universal look and feel.
- *Support for pen and voice input* would allow to make use of advanced technology and extend the range of interaction styles.

2.4 High Data Volume

One common characteristic of all the media types discussed before is the requirement to support the storage and management of huge amounts of data. Figure 1 shows the memory consumption for several common media types and Figure 2 the bottleneck of storage and transfer rates in a workstation-based environment. In the case of time-independent data (text, images, graphics) at least no serious problems in terms of processing speed are imposed on storage devices and networks. As soon as a system has to deal with time-dependent data the problem of very high data volume becomes critical because of the additional timing constraints imposed on the processing of time-dependent data. This may influence the design of operating systems, networks, and hardware.

Media Type	**Format**	**Volume**	**Transfer Rate**
text	ASCII	1 MB/500 pages	2 KB/page
b/w image	G3/4-FAX	32 MB/500 images	64 KB/page
color image	GIF, TIFF; JPEG [1]	1.6 GB/500 images 0.2 GB/500 images	3.2 MB/image 0.4 MB/image
speech	μ-law, linear; ADPCM, MPEG audio [1]	2.4 MB/5 min. 0.6 MB, 0.2 MB/5 min.	8 KB/sec.
CD-music	CD	52.8 MB/5 min.	176 KB/sec.
consumer video	PAL	6.6 GB/5 min.	22 MB/sec.
high quality video	HDTV	33 GB/5 min.	110 MB/sec.

Fig. 1. Media types, formats and resource consumption

When dealing with such high data volume it may be more efficient to operate on abstractions of the data instead of the data itself. But the final presentation of

[1] compression formats

the media will always rely on the original data. In such a case a form of dynamic data management is needed in order to bridge the gap between resources available and resources needed. Techniques employed in this context are pipelining data, prefetching data, splitting the presentation operation into several processes etc.

Source	**Volume**	**Transfer Rate (BYTE/sec.)**
main memory	64 MB	ca. 100 MB
floppy disc	1.5 MB	ca. 50KB
MO-disc	300 MB/side	620 KB
CD-ROM	644 MB	150/600 KB
magnetical disc	1/2 GB	5/10 MB
RAID system	30 GB	
MO disc jukebox	50 GB	
tape change system	600 GB	
telephon		ca. 2.5 KB
S-ISDN		2 * 8 KB
Ethernet (typ./1:1)		0.12/1.2 MB
FDDI		2.4/12 MB
ATM		4/20 MB

Fig. 2. storage and channel transfer rates

2.5 Transport of Multimedia Data

Multimedia applications usually require the storage of multimedia data at a server site and the consumption of the data (playback, presentation, or general manipulation) at a client site. Traditional networking environments like LAN do not meet the requirements with respect to high data rate, stream-oriented highly bursty traffic patterns, less rigorous reliability, latency, or synchronized transmission. Specific solutions are needed, e.g. with respect to traffic channels (e.g. separating multimedia traffic from regular network traffic according to priority schemes) [Cri93] and quality of service [LG90].

2.6 Integrated Multimedia System

To justify the notion *Integrated Multimedia System* a system should meet the following demands [HS91, MW91, RSSS90]:

- *Combination of Media:* The system should offer the free combination of different media to one multimedia object. This applies both to the temporal and spatial dimension of multimedia presentations.

- *Integration of Media:* The composition of the media leads to one single object. Manipulation operations on an object should be available in such a way that the work on and presentation of it seems as if it was one new medium.
- *Independent Access to Media and their Parts:* At the same time it must be possible to manipulate single media independently from the whole composition. This applies as well for the parts of a single medium.
- *Translation of Media:* Translation in this context stands for the conversion from one medium into another. Representatives are the conversion of text into audio by means of speech generation, and the conversion of pictures into text with optical character recognition.

3 Multimedia Database Systems

In this chapter, we emphasize the current issues in the development of multimedia database systems and some solutions achieved so far. We describe four basic mechanisms which can be used in current DBMSs to manage multimedia information. Nevertheless, adequate DBMS support for multimedia systems requires more than just writing a database schema. We introduce four notions to classify multimedia systems: hybrid, structural, behavioral, and distributed multimedia system. A "true" MM-DBMS should support all notions to some degree. We describe in more detail the constituents of a MM-DBMS: multimedia modeling, content-based retrieval, continuous storage management, and a distributed architecture.

3.1 Storing Multimedia Information in Database Systems

Multimedia information can be stored in database systems by different mechanisms [KMMW93, RLM$^+$93]. We describe four basic types which may appear in current DBMSs alternatively or jointly.

- *External References:* Using this mechanism, a database contains references to the original multimedia data. The references may be file names or other identifiers to locate the data in an open environment [ISO91]. In addition to the references the database contains descriptive data modeled as additional attributes or relationships. For example, attributes of a video stream may be its length, the applicable output device, its compression format, and a textual description of the content. Obviously, the database system cannot provide support for the original multimedia data.
- *Long Fields:* A long field or *BLOB* (binary large object) usually can store up to several GBytes of data and can be used as a domain for attributes. Essentially, a long field can be accessed partially. The DBMS provides full support for data stored in long fields. Nevertheless, contents is still uninterpreted, structuring takes place at the (low) level of bytes, and the functions offered to operate on the data are generic only.

– *Using External Functions:* Some database systems allow to call external functions for processing data stored in the database. The limitations on the data manipulation language such as SQL which are not universal programming languages are the reason for introducing this mechanism. Multi-user support and authorization can be applied for the *access* to external functions but their execution cannot be controlled by the DBMS. For example, if an audio is delivered to an audio device using an external function exclusive access cannot be guaranteed by the DBMS. However, external functions are often very useful to reuse existing algorithms and tools in the context of multimedia presentation and capture.
– *Extensible or Object-Oriented System:* Extensible systems allow the programmer to define (at least) abstract data types and to refer to them in applications. Especially, the object-oriented approach supports advanced modeling by building up datatype hierarchies and defining relationships between instances of classes [KAN93]. In some systems, specific indexing mechanisms can be established for user-defined datatypes [SK91] thereby allowing extensibility *within* the system. Object-oriented systems offer the most suitable support for multimedia information but still lack some features such as supporting time-dependent data, user interaction, and content-based query and retrieval techniques.

Most DBMSs support client access to a database which reside at a remote server. Thus, distribution is supported if multimedia information is stored in long fields or an extensible/object-oriented system is used. However, specific transport protocols for streams are not available. Figure 3 summarizes our discussion [RLM+93]).

3.2 The Notions for Multimedia Database Management Systems

A DBMS can be classified as a multimedia DBMS if - roughly speaking - all entries in Figure 3 contain a "Yes". However, several levels of multimedia support can be distinguished with respect to usage of analog devices, structuring features, handling of time-dependencies, and the degree of distribution. Our classification follows the historical developments of multimedia database systems. In our opinion, a current multimedia DBMS should support all levels to some degree. The need to use analog components (hybrid systems) will decrease with the availability of efficient digital components.

– *Hybrid Multimedia System:* A system is called hybrid if analog components as well as digital can be used together [HS91]. Typical hybrid multimedia systems use analog storage devices like VCR and optical disks which are controlled via the digital components and/ or audio and video data are transmitted via analog cables. Because analog components guarantee required data transfer rates, high quality systems can be build. However, specific hardware

[2] limited for database objects only

DBMS Property	External References	Long Fields	External Functions	Extensible/ Object-Orient.
Persistency	No	Yes	No	Yes
Data Independence	No	No	No	Yes
Indexing	No	No	No	Yes, limited
Object Buffering	No	Yes	Yes [2]	Yes
Multi-user Support	No	Yes	Yes [2]	Yes
Recovery	No	Yes	Yes [2]	Yes
Authorization	No	Yes	Yes [2]	Yes
MM Data Types	No	No	No	Yes
Time Dependency	No	No	No	No
Interactivity				
(i) Presentation	No	No	Yes	Yes
(ii) Control	No	No	No	No
High Data Volume	No	Yes	No	Yes
Distribution	No	Mostly Yes	No	Mostly Yes
Streams	No	No	No	No

Fig. 3. Support for multimedia information by current DBMS mechanisms

must be installed and tape technology may result in long start-up delays. Expensive conversion between analog and digital formats is necessary if operations manipulate data or if analog data is stored on a digital storage device.

- *Structural Multimedia System:* As introduced in [Dit86] a system is called structural if composite objects are supported. This includes datatype constructors (e.g. array, set, list) and generic operators to access composite objects in their entirety as well as parts (elements) of structured attributes. Operations offered are generic reads and writes on datatypes, combined with object creation and deletion operations. In [Loc88], a structural multimedia DBMS is preferred because it should not "preempt the semantics of the served systems". Multimedia information like graphics and text require adequate modeling of structural properties. Audio and video can be seen as an array of samples or frames, respectively. A synchronized data stream can be modeled as a list of the single media data objects combined with a description of its timing requirements.
- *Behavioral Multimedia System:* New specific behavior for multimedia data results from the need to present them to the user under particular timing requirements. The same holds for capture of multimedia information but for practical reasons this situation may be handled differently. If data of an video conference, for example, has to be recorded data must not be lost. In the case of a presentation, degradation of QoS parameters may be tolerated temporarily. Thus, a multimedia system is called behavioral if at least time-dependent and synchronized presentation of multimedia data is supported.

Note, a behavioral object-oriented system [Dit86] models the behavior of conventional datatypes but not necessarily the behavior of time-dependent data. However, behavior of multimedia data cannot be modeled without a notion for the behavior of conventional data.

In behavioral systems, we can differentiate levels of assistance for synchronization (see 4.4). *Fine-grained* synchronization is required at the user's workstation for presentation purpose. *Coarse-grained* synchronization is appropriate if less restrictive timing requirements are allowed. A usual technique is buffering of data.

– *Distributed Multimedia System:* It follows from the nature of multimedia data that they have to be viewed or heard by the user at his workstation. Thus, distribution of multimedia data must be enabled by multimedia systems if multi-user access is supported. Transport of multimedia data between the nodes of a distributed system must be supported by the notion of a data stream. We can differentiate between several architectures with respect to object availability. In a *client/server* system objects are transparently transferred from the server to the client. Implementations may or may not support buffering (caching) at the client. Clients can access objects only via the server. In a (fully) distributed system (unrestricted) *object migration* over all participating nodes is supported, allowing access from every node in the system. This mechanism supports applications where users cooperate or data sources at different sites have to be accessed. The advantage of a client/server architecture for DBMSs is its easier implementation and mangement overhead through centralized concurrency control, recovery, and authorization.

An example of a structural multimedia DBMS is the ORION DBMS [WK87]. Multimedia data, their presentation and storage can be modeled, but none of the other characteristics of multimedia systems are supported by the DBMS. The MOSS DBMS is a structural system, too, which includes a comfortable set datatype constructor [KMMW94]. The V^3 Video-Server is a multimedia application based on the VODAK DBMS [RM93]. The object-orientation supports the structural aspects of multimedia data. The transparent access of analog video clips stored on an optical disk classifies the video server as hybrid. Additionally, the video clips are transparently distributed to clients. Because coarse-grained synchronization was not supported by the VODAK DBMS, fine-grained synchronization at the client was realized by prefetching all frames of a requested video clip.

3.3 Multimedia Data Modeling

The relevant quality of service (QoS) parameters from the perspective of the database are the average delay at the beginning of a presentation, the speed ratio between desired and actual speed, and the utilization of stored and presented data. The parameters themselves are not independent of each other, e.g. when presenting a video it might be appropriate to fix the speed ratio and to change the utilization in order to overcome overload situations.

Representation of alphanumeric data is not a problem at all. Formatting problems are mostly settled and supported by default by operating systems and programming languages. The basic datatypes are not adequate to reflect the structure of multimedia data. New built-in datatypes like *bitmap* and *audio* and a notion of *stream* for presentation and capture purposes are needed. These built-in datatypes need to have media-specific operations. Spatial data should be accessible by coordinates (e.g. point, area, volume), time-dependent by their time axis (e.g. in 1/10.000 seconds). Manipulation operations may be available for different media but need different implementations. For example, attack and decay are needed for audio and video datatype(s) but have to be implemented differently. The object-oriented concept of overloading and of overriding of operations can be applied here.

In addition to the datatypes, type constructors which allow to deal with the temporal relationships are very useful. For example, a type constructor which models the synchronized combination of a video (e.g. sequence of bitmaps) and an audio (represented as an audio stream) can be used to create the representation of a video clip including its sound-track.

The introduction of appropriate datatypes and type constructors is complicated by the use of compression techniques. There is not just a single compression technique applicable for all multimedia data and all types of applications. Hence, supporting different compression techniques results in different representation formats which underly the datatypes and type constructors for continuous data. The system must provide for a modular and efficient representation of these standards and should make them transparent to the user.

3.4 Content-Based Retrieval

Retrieval in multimedia databases must include the type of queries known from the field of traditional databases as well as retrieval functionality (such as full text search) known from the field of information retrieval. In the case of videos e.g. content based search means to be able to search for a specific fragment of a video which starts with a given scene, or includes given objects. In the case of audio, one might want to be able to retrieve all those audios which are associated to some given topic. Such a retrieval operator may include interactive input needed to process the query. For example, the user's sketch of the shape of cars he is interested in may be used as parameter of a query. Retrieval/query operators should allow for the composition of new media as the result of a request. An example is the retrieval of a particular video and a particular audio (in a specific language) which returns to the user a synchronized multimedia object which can be played lip-synchronously.

Content-based retrieval may depend on the availability of rich metadata or metaknowledge about the original multimedia data. The metadata can be based on additional knowledge incorporating the semantics of the data and its intended usage in a particular application. Or it might be (semi-)automatically derived from the original data by employing specific analization techniques.

3.5 Continuous Storage Mangement

In order to provide timely delivery continuous data streams may be directed from the storage components (see 1.1) to the consuming component (viewer, application) bypassing other layers of the multimedia database system. This avoids additional overhead but does not allow any further processing (selection of portions, scaling, etc.) of the data by the database system. In this case the storage component must offer some high level delivery protocol for the consuming components. The protocol itself can be based on network transport protocols which meet the requirements of transmitting data over networks in terms of quality of service parameters.

3.6 Architecture

Every system that provides special support for continuous media must deal with the central aspects of media storage, transport, manipulation, and presentation. Due to the high expense these aspects should be efficiently managed by one integrated system. An OO-DBMS already allows the modeling of large amounts of complex structured data for several concurrent users and applications. The ability to store data together with the appropriate manipulation and presentation methods is of special interest for multimedia because media data are a lot more dependent from these methods than conventional data.

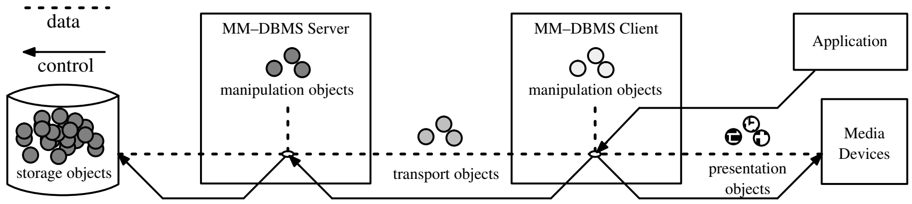

Fig. 4. General architecture of a multimedia DBMS

Figure 4 shows a multimedia application that uses the services of the DBMS to retrieve multimedia objects from the database, to manipulate them, to transport them over the network and finally to present them at the user's workstation. The representation of the objects may depend on the task to be executed on the objects. Objects may be converted between specific representations for storage, transport, manipulation, and presentation.

A transport protocol that implements the continuous flow of data along with a mechanism for continuous control is of central importance for an efficient management of presentation and capture functionalities throughout the whole system. Usual applications never get in touch with the data stream itself. They access their local client of the MM-DBMS to initiate a presentation of data from a central database at the local workstation. The MM-DBMS then retrieves the media data in the database of the server and sets up a continuous data connection from the the server through the network to the client. By means of this connection the data are transferred to the local presentation device(s). The data connection and a corresponding control connection are working in parallel to the other components of the distributed DBMS and the application. One data connection can consist of several channels to support the presentation and capture of a set of media.

4 The Multimedia Database Management System AMOS

So far the properties of an OO-DBMS like `VODAK` are useful for a multimedia system. Our implementation of a video-server [RM93] as a standard `VODAK` application, however, has shown that they are not sufficient. Current OO-DBMSs do not offer support for timing and continuous data transport. To capture the continuous and time-based character of media such as audio and video special solutions for the problems of continuous transport, presentation and control are needed.

Schemas are modeled in the `VODAK` modeling language (`VML`) [KAN93]. Application programs are written `VML`or in C++ using the `VODAK` C++ programming interface. Within a `VML`schema metaclasses, application classes and object types are specified. The inheritance hierarchy of types together with the transformation of types to classes and metaclasses are defined here, too. Every VML application program can use the navigating access facilities of VODAK, which consists of methods to access all instances of a class and the dereferencing of object identifiers. The query and update language of the `VODAK`DBMS is based on a declarative query approach. It differs from SQL mainly by allowing method calls within a query statement.

In this section, the developments achieved so far for the AMOS prototype are observed. The design of the datatype `AUDIO`, the modeling of meta information, multimedia presentations, the development of an object manager for continuous objects, and the integration of the information retrieval system INQUERY in our system are described.

4.1 The AMOS Architecture

In a client/server environment data objects can be partitioned between server and client. In the general architecture of a multimedia DBMS (Fig. 4), full modeling capabilities on both sides are presumed. We decided, however, for ease of implementation and for re-implementation on other platforms to assume an environment with built-in data types. Hence, persistent objects are created only at the server. This restriction is necessary to prevent conflicts and inconsistencies produced by uncoordinated changes on local replicas of the media object. In addition, it may be impossible to copy complete audio objects to the clients due to their limited storage capacity. As a consequence of this partition calls to manipulation methods are always sent to the server for execution. Changes applied to the audio data become visible to concurrent users of the database immediately after the completion of the transaction.

4.2 The Datatype `AUDIO`

The structure of audio data together with methods for their manipulation could be implemented on the level of objecttypes where it could use the available modeling techniques of the `VODAK` modeling language VML. This approach would result in the fact that the underlying system would have no information about the special characteristics of audio as a multimedia type. Especially the need for parallel, continuous I/O and transport as well as the necessity of interactive control could not be expressed by means of the standard modeling capabilities. In addition, the usage of audio should be elementary and simple for the user. Therefore we decided to implement a prototypical audio datatype as the first step towards a complete multimedia DBMS [LR94].

The decision upon the storage format for audio data was based on both our experience with compression techniques and the typical applications we expected:

- *Data Compression vs. Raw Data:* With regard to the storage format of the audio data we had to decide whether any of the currently available compression methods are suitable for our purposes. Compression raises a general problem because manipulation methods usually need access to uncompressed data. Manipulation of compressed audio always means to decode the compressed data, apply the changes and encode it again. For applications like the production of multimedia documents for which editing is one of the main tasks, this procedure causes more than performance problems. Even with the most advanced compression techniques repeated decompression, manipulation, and compression cycles introduce an audible loss of quality due to the effect that minor deteriorations are accumulated. As a consequence of our experiences with the exemplary compression methods and because of the performance and deterioration problems we decided not to use compression for the time being.
- *High Quality Format vs. Parameterized Format:* For all operations manipulating audio data one common high quality format would be a convenient basis. Converting low resolution digital audio to a high definition format, however, would increase space consumption by a factor of at least 20. Further, format conversion may generate false information resulting from the algorithms that generate samples by means of interpolation that were not present in the original. Without careful signal processing this effect may be

audible as noise. In addition, frequent format changes are unlikely to happen with typical applications as they are expected to choose a format suitable to their demands in advance. The considerations on available space, computing power, and expected usage lead to the concept supporting a parametrized format. In this context parametrization means to allow any sampling rate and sample sizes of 8, 16 or 32 bits. Linear encoding is supported for all sample sizes. 8 bit samples may be μ-law encoded alternatively. To preserve compatibility between different parametrizations adequate conversions are available on demand.

According to the considerations on data formats and bearing our area of application in mind, we implemented the datatype AUDIO including a set of built-in operations. These operations can be divided into the following groups:

- *Import/Export from/to Files:* We support several common file formats including uncompressed 8- and 16-bit mono and stereo audio data.
- *Recording and Playback:* In contrast to normal operations or method-calls in the `VODAK` environment calls of these operations are executed asynchronously. This allows calls to return before playing or recording is completed. The interface of these operations hides an interrupt-controlled mechanism that switches between two buffers and enables continuous operation. An inquiry operation can be called to detect whether one of the two buffers is free. In this case play or record return immediately. Otherwise these operations have to wait. This dual buffer mechanism allows to call `play` asynchronously once again while the audio from the previous call is still being played [LR94].
- *Inquiry of Attributes:* By means of the operations of this group information about sampling rates and sample resolutions can be obtained. Furthermore the operations allow access to status and current position of ongoing play and record operations. Status and position information are important for the continuous transport mechanism and the synchronization with other media.
- *Audio Manipulation:* Supported manipulations are cut and paste operations, operations to change the volume persistently, to mix several sources into one destination, and to convert between different sample rates and resolutions. The implementation of a low-pass filter operation was necessary to improve the quality of the sampling rate conversion. An attack/decay mechanism and a method for dynamic compression have been implemented as two more complex operations based on volume manipulation. The attack/decay mechanism allows to produce effects such as fade-in and fade-out. Dynamic compression provides selective amplification of passages with low volume and therefore reduces the dynamic range of the audio signal.

4.3 Modeling Meta Information

The rich semantics carried by multimedia data can be utilized be content search, which is a very difficult problem as can be seen from the information (text) retrieval systems. The problems of content-based search lie outside the scope of database research and are difficult to solve. The task of a MM-DBMS is to provide the framework for doing content search. The organization of the data must allow for operations to be added easily. A clear subdivision into different categories of data is necessary: some data are useful for content queries, other data are only needed for internal use such as presentation [TR93]. This implies the need for a meta organization of the data. The following classification of metadata for multimedia objects can be made [BR94]:

- *Metadata for the Representation of Media Types:* This includes format, coding and the compression techniques that have been applied. Either a current name for format, coding etc. may be given, or it may be explicitly described. For example, the datatype `AUDIO` is described by the number of samples per second, number of channels, and the coding in which it has been recorded. From our point of view, certain attributes of textual document components not reflecting the content also fall into this category. An example is an attribute `language` bearing the language a textual component is written in.
- *Content-Descriptive Metadata:* These metadata are determined intellectually or by means of semi-automatic or automatic methods. In the last two cases, these methods are media-type-specific. Examples of content-descriptive metadata are a list of persons or institutions having some relation to a particular multimedia document's content.
- *Metadata for Content Classification:* The distinction between content-descriptive and technical metadata for simple digital media types is well-known (cf. [MW91]). Furthermore, however, while content-descriptive metadata reflect an object's or an object component's content, metadata for content classification are additional information that can be derived from the object's content. For instance, metainformation such as the level of expertise in the field required by the reader of a text falls into this category. To our knowledge, coming up with algorithms how that kind of metainformation can be derived from the document content automatically or at least semi-automatically is an open research issue.
- *Metadata for Composition and Relationships:* The relationships between objects have certain characteristics. Composition-specific metadata are knowledge about these characteristics. As opposed to some other kinds of metadata, this one makes sense only for structured multimedia objects and not for simple media types. For instance, the direct content elements of a document component may be ordered, as with SGML documents [ISO86], or not, as with diverse hypertext models. Using data modeling terminology, the different semantics of these instances of the partOf-relationship is reflected in the operations that would be provided by the corresponding modeling primitives. In the first case a method `getNextComposite` returning the composite that follows the target object makes sense. This method, however, would not have a counterpart in case of unordered composites.
- *Metadata for Location:* Multimedia objects are not inclined to be duplicated and distributed, as compared to conventional ones. Rather they are accessed

by the consumer on demand. This facilitates continuous modification of documents' content without that the object actually becomes another one. It is a prerequisite, however, that the multimedia documents can always be localized unambiguously [ISO91].

The kinds of metadata that have been mentioned so far relate to individual multimedia objects. Additionally, there are metadata for collections of multimedia objects. Here, we give an example of such metadata which will be referred to as *statistical metadata.* Consider the case that multimedia documents are stored within a database. Metainformation that may be relevant in this context is the frequency of documents with certain characteristics.

4.4 Modeling Multimedia Presentations

An application on the MM-DBMS-Server can compose a presentation on a high definition level, e.g. HyTime [AK94, ISO92]. We map this high-level description to a simple representation that contains all information that concerns the media and their occurrence in the course of the presentation and use it at the client's site for the actual presentation.

Our system allows the free composition of different media to a new multimedia product, a multimedia presentation. Any combination of both *continouus* media such as audio, video, and text as well as *non-continuous* such as picture can be arranged in one multimedia presentation. This calls for a modeling of a presentation that includes defined temporal time dependencies between the media, defined time intervals in which media are presented to the user as well as media specific characteristics such as the initial playback volume of an audio.

Modeling: The modeling of multimedia presentations has to take the differents aspects important for the presentation into account and has to meet the demands of the users of a multimedia presentation. The essential reflections of the modeling and our solutions can be summarized as follows:

- *Spatial and Temporal Composition:* Our way to describe a presentation is suitable for the definition, storage, and playback of a complex multimedia presentation at the client. The description reflects both all possible temporal relationships between the media, e.g. played in parallel, and necessary media specific information. Besides the description of time in a multimedia presentation a possible spatial position of two-dimensional media on the screen and the overlapping of two-dimensional media are modeled.
- *Time Line:* The selected representation mirrors the entire temporal course of the multimedia presentation. Our solution to keep track of a presentation is to store information only about changes during a presentation. Changes take place in a presentation at certain moments (*events*) on a time line: for each medium in a presentation the start, the end, and every alteration event are stored in a script-like fashion. For each event the time of occurrence is stored by means of (*relative time intervals*) between the events on the

time line. The sequence of serial and parallel events during a multimedia presentation including each the necessary presentation parameters is kept in the flat structure of a list, implemented in VML. An interesting difference between our solution and the conventional time line is that media without predefined end can be represented in our system. An example of the necessity of the latter is a help window in the presentation. The end of its presentation is determined interactively.
- *Interaction Capabilities:* One of the main features of a presentation on the client's site is the user interaction in the course of a presentation. As indicated above interactions are treated and modeled as normal media. The standard interaction resemble the typical interactions on the control panel of a video recorder. Over and above that complex interactions such as *selection* are currently under construction.
- *Coarse and Fine Synchronization:* The modeling of multimedia presentations raises the question of how to represent its synchronization features. Timing requirements are subdivided in fine and coarse synchronization. The coarse synchronization ensures that the time line representation of the presentation is put into action. Coarse synchronization relies on the mere schedule of the presentation stored in the description. The fine synchronization, however, obeys the maximum permissible deviation from refence media. For each single event on the time line one or more reference media to synchronize with are specified in the script-like representation as well as the corresponding deviation limit.
- *Presentation Parameters:* Initial settings such as playback volume for an audio, the playback speed of a video and the like are modeled. The exploitation of the knowledge about the presentation environment at the client's site flows into the script-like representation at the server when it is generated. This improves the quality of the actual presentation at the client.

Presentation: The script-like description of the presentation is transferred from the MM-DBMS-Server to the client, is interpreted there, and the presentation is showed to the user as desired. The interpreting component at the client manages the preparation, startup and termination of the single media presentations that belong to a complex multimedia presentation. The implementation of the interpreter of the script-like representation of a multimedia presentation is currently under construction.

4.5 The Continuous Object Manager

The continuous object manager frees the applications from considering time-dependency during media capture and presentation. Continuous object management functionality is categorized into object handling, direct access, and buffer resource management. Additionally, it has been found that traditional communication protocols, e.g. TCP/IP or OSI-like protocols are not sufficient for real-time requirements of multimedia applications [LG90, Nic90]. Therefore, the integration of a multimedia transport protocol is planned.

The client/server distribution of the AMOS system is constructed in such a way that a distributed database buffer is maintained by the continuous object managers on the server and the client. The support of interactions for continuous data leads to a new understanding of buffer management strategies. The well-known statical buffer preloading and replacement strategies (e.g. most recently used etc.) are substituted by more elaborate algorithms which consider the actual structure and behavior of continuous data streams as discussed in [R$^+$94]. The primary idea is described by an example of the presentation of an M-JPEG (Motion JPEG) video clip (Figure 5).

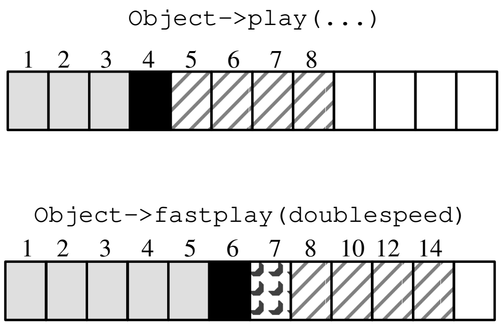

Fig. 5. Continuous Object Management

At the beginning of the presentation the method call `play()` is sent to the respective object `Object`. The continuous object manager initializes its buffer by preloading continuously the JPEG-frames which are needed to best support the presentation state `play`. In our example in figure 5, frame 4 is being presented, 1 to 3 were already displayed and frames 5 to 8 are preloaded. While consuming frame 4, the user changes the presentation status from `play` to `fastplay(doublespeed)`. The operation may be implemented so that every second frame is dropped. Hence, the new preloaded frames are 10, 12 and 14. Frame 7 is no longer needed. When the presentation direction is changed, the continuous object manager can use the same strategies by preloading “on the left”. A state transition from `fastplay` to `play` is realized by frame stuffing of eventual missing frames.

A simple and sufficient replacement strategy is as follows: replace frames which are farthest away from the actual presentation point.

Other relevant questions of continuous object management are the intra-media synchronization of different media streams, how the buffer resource is distributed over several multimedia presentations and how and which scaling [D$^+$93] or adaptation strategies on the client and the server side can be considered.

4.6 Content-Based Text Retrieval

Whereas DBMSs are particularly suited to handle structured information in multi-user environments, complement information-retrieval systems (IRSs) provide content-oriented retrieval capabilities [SM83]. Result of an IRS query are a set of documents and their (probabilistic) *relevance* for the query. We have integrated the IR system INQUERY with the object-oriented DBMS VODAK [VAB95]. Combining structural queries with IRS queries leads to non-trivial questions with regard to retrieval semantics and query processing. Further, the VODAK DBMS supports the management of user-definable typed document structures according to SGML and the hypertext features of HyTime [ABH94, BA94, ISO86, ISO92]. In the sequel, we give some examples of possible queries in this system:

- *"Select the authors of all chapters whose introduction is about multimedia databases"*
- *"Select the titles of all sections on the information highway"*

Naturally, such queries require that the document-type definition contains element types 'chapter', 'introduction', and 'title', and that chapters have an attribute 'author', and that they may have an introduction.

We have coupled the DBMS to the IRS *loosely*. The DBMS facilitates a mapping between logical entities in the database and IRS-documents. The database schema reflects which are the logical units from the application's point of view. We want to remain open for arbitrary document components, not only entire documents, being these logical units. The IRS administer flat text only. Besides that, the result of IRS queries shall always be complete IRS documents. Applications access the DBMS which makes the DBMS the control component of the IRS. Thus, modifying any of the existing systems is not necessary. Queries issued by the application are expressed in the database query language making use of query-processing mechanisms, i.e. analyzing, evaluating and optimizing queries. Formulating complex queries is easy using the database query language. Other database features are "for free". The most important result is that query results can be given which contain the relevance of specifically selected document parts.

5 A Sample Application

In this section, we show how our system can be used within a sample application. Other current applications are a multimedia calendar of events (CoE) [R+94, TR94] and an interactive audiotool for an audio/image database (AAT) [LR94].

This application supports the engineering process such that dependencies between different multimedia specification and report documents can be managed automatically and global consistency is ensured. The $\mathcal{M}uSE$project [3] aims at the integrated system support of the systems engineering process such that dependencies between different specification and documentation documents can be managed automatically and global consistency can be ensured [DGJ+94] .

$\mathcal{M}uSE$ covers various phases of the systems engineering process including design, verification, animation, and simulation, and follows the concurrent engineering paradigm.

In the $\mathcal{M}uSE$-environment the information resulting from the design process, verification of specifications, system simulation and animation, and testing is stored in the underlying object-oriented and multimedia database management system. This includes alphanumeric data as well as graphics, images, audio and video annotations which may originate from simulation and animation results. The system allows for the storage, retrieval and manipulation of highly structured information like 3D-data, part structures, and multimedia and hypertext documents in a multiuser environment which need to be supported by the underlying database management system.

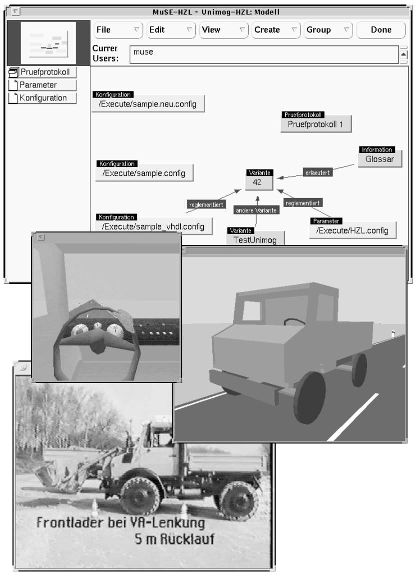

Fig. 6. Screendump of the $\mathcal{M}uSE$ environment

The $\mathcal{M}uSE$ prototype uses hypermedia concepts to organize the documents of the system development process. The complete system model is represented as a hypernetwork containing the different specifications. The hyperstructure is visualized via a hypermedia authoring environment which provides the desktop

[3] $\mathcal{M}uSE$ is the acronym for a project entitled *Multimedia Systems Engineering*. The project is a joint effort of groups at the Technical University of Darmstadt, GMD-IPSI and FhG-IGD, Darmstadt. The project is sponsored by the *Deutsche Forschungsgesellschaft* DFG, grant numbers He 1170/5-1 and He 1170/5-2.

for the whole $\mathcal{M}uSE$ environment. Figure 6 shows a screendump of the $\mathcal{M}uSE$ environment.

6 Conclusions

In this paper we have investigated the functionality needed to support advanced multimedia applications. We have illustrated how OO-DBMS technology can be applied and what need to be modified and/or extended internally to support such applications. We have compared our approach to other related work and have pointed out differences and advantages of the various systems. However, we do not claim to have found a complete solution for the many features and properties required in such a MM-DBMS. An important open issue remains what the adequate level of application support is.

We also feel that more work has to go into effective storage models and models on how to store but especially retrieve multimedia information. Traditional query based retrieval does not seem feasible as it is nearly impossible to precisely describe, via attributes, the content of a video or image. Imprecise queries as they have been handled in the field of information retrieval will have to be embedded into the database management system. Again how much is integrated and how much is built on top is to be investigated further.

In contrast to the prominent rôle of object-oriented DBMS in this article results of other DBMS research areas have not been applied. Nevertheless, we are sure that the research that is and will be executed in *active DBMS*, *real-time DBMS*, and *distributed* as well as *heterogeneous DBMS* will contribute to solving the problems of MM-DBMS's. The history of scientific research shows that some problems are re-solved by different areas. For example, scheduling of processes takes place in operating systems as well as in DBMS's. We strongly hope that developments in the area of multimedia DBMSs, *multimedia operating systems* (including file systems), and *multimedia network protocols* will benefit from each other. Maybe, that—by the fascinating domain of handling multimedia information—these areas will be tied together.

Acknowledgements: We would like to acknowledge the work of our colleagues *Wolfgang Klas*, *Karl Aberer*, *Klemens Böhm*, *Frank Moser*, *Heiko Thimm*, and *Marc Volz* contributing to the work on the AMOS system. We also thank the $\mathcal{M}uSE$ team for providing material, and our many students working with us on the development of the system components. Especially, the participation of *Susanne Boll* in preparing this paper was very helpful.

References

[A+92] D. P. Anderson et al. A file system for continuous media. *ACM Transactions on Computer Systems*, 10(4):311–337, November 1992.

[ABH94] K. Aberer, K. Böhm, and C. Hüser. The prospects of publishing using advanced database concepts. In *Proc. of the International Conference on Electronic Publishing, Document Manipulation, and Typography, EP94, Darmstadt, Germany*, pages 469–480. John Wiley & Sons, Ltd., 1994.

[AK92] K. Aberer and W. Klas. *The Impact of Multimedia Data on Database Management Systems*. Technical Report TR-92-065. International Computer Science Institute (ICSI), Berkeley, CA, USA, 1992.

[AK94] K. Aberer and W. Klas. Supporting temporal multimedia operations in object-oriented database systems. In *Proceedings of the IEEE International Conference on Multimedia Computing and Systems, Boston, USA*, May 1994.

[B+90] P. B. Berra et al. Architecture for distributed multimedia databasesystems. *Computer Communications*, 13(4):217–231, May 1990.

[BA94] K. Böhm and K. Aberer. An object-oriented database application for hytime document storage. In *Proceedings of the Conference on Information and Knowledge Management (CIKM94)*. Gaithersburg, MD, December 1994.

[BAH94] K. Böhm, K. Aberer, and C. Hüser. Introducing D-STREAT - The Impact of Advanced Database Technology on SGML Document Storage. $\langle$*TAG*$\rangle$, 7(2):1–4, February 1994.

[BR94] K. Böhm and T.C. Rakow. Metadata for multimedia documents. In *SIGMOD Record (Special Issue on Meta-data for Digital Media)*, number 4 in SIGMOD Record. ACM, December 1994.

[CHT86] S. Christodoulakis, F. Ho, and M. Theodoridou. The multimedia object presentation manager of minos: A symmetric approach. In *Proc. Int. Conf. on Management of Data, Washington*, pages 295–310, 1986.

[Cri93] S. M. Crimmins. Analysis of video conferencing on a token ring local area network. In *Proceedings of the ACM Conference on Multimedia 93*, pages 301–310, 1993.

[D+93] L. Delgrossi et al. Media Scaling for Audiovisual Communication for the Heidelberg Transport System. In *Proc. ACM Multimedia Conf.*, 1993.

[DGJ+94] M. Deegener, G. Große, W. John, B. Kühnapfel, M. Löhr, and H. Wirth. Rapid Prototyping with MuSE. *International Symposium on Automotive Technology and Automation, Dedicated Conference on Mechatronics*, 1994.

[Dit86] K. R. Dittrich. Object-oriented database systems: The notion and the issues (extended abstract). In *K. Dittrich and U. Dayal (Eds.): Proc. Int. Workshop on Object-Oriented Database Systems*, pages 2–4. IEEE CS Press, 1986.

[EF94] J. L. Encarnação and J. D. Foley, editors. *Multimedia*. Springer Berlin, 1994.

[Fur94] B. Furht. Multimedia Systems: An Overview. *IEEE MultiMedia*, 1(1):47–59, 1994.

[GC92] J. Gemmell and C. Christodoulakis. Principles of Delay-Sensitive Multimedia Data Storage and Retrieval. *ACM Transactions on Information Systems*, 10(1), January 1992.

[Gro94] W.I. Grosky. Multimedia Information Systems. *IEEE MultiMedia*, 1(1):47–59, 1994.

[HS91] R.G. Herrtwich and R. Steinmetz. *Towards Integrated Multimedia Systems: Why and How*. Technical Report 43.9101. IBM, march 1991.

[ISO84] ISO. *PHIGS - Programmers Hierarchical Interface to Graphics Systems*, 1984. ISO/TC97/SC5/WG2/N305.

[ISO86] ISO. *Information processing - Text and Office Systems - Standard Generalized Markup Language (SGML)*, 1986. ISO-IS 8879.

[ISO91] ISO/IEC. *Information Technologie - Text and office systems - Distributed Office Applications Model (DOAM), Part 2: Distinguished-object-reference and associated procedures*, 1991. ISO/IEC 10031.

[ISO92] ISO. *Information Technology - Hypermedia/Time-based Structuring Language (HyTime)*, 1992. ISO/IEC IS 10744.

[KAN93] W. Klas, K. Aberer, and E. Neuhold. Object-Oriented Modeling for Hypermedia Systems using the VODAK Modelling Language (VML). In *Object-Oriented Database Management Systems, NATO ASI Series.* Springer Verlag Berlin/Heidelberg, August 1993.

[KMMW93] R. Käckenhoff, D. Merten, and K. Meyer-Wegener. Eine vergleichende Untersuchung der Speicherungsformen für multimediale Datenobjekte. In *W. Stucky and A. Oberweis (Hrsg.): Datenbanksysteme in Büro, Technik und Wissenschaft*, pages 164–180. Springer Verlag Berlin, 1993.

[KMMW94] R. Käckenhoff, D. Merten, and K. Meyer-Wegener. MOSS as a Multimedia Object Server. In *Proceedings of the 2nd International Workshop on Advanced Teleservices and high Speed Communication Architectures, LNCS*, pages 413–425. Springer Verlag, 1994.

[KNS90] W. Klas, E. J. Neuhold, and M. Schrefl. Using an Object-Oriented Approach to Model Multimedia Data. *Computer Communications, Special Issue on Multimedia Systems*, 13(4):204–216, May 1990.

[LG90] T. D. C. Little and A. Ghafoor. Network Considerations for Distributed Multimedia Object Composition and Communication. *IEEE Network*, 4(6):32–49, November 1990.

[Loc88] P. C. Lockemann. *Multimedia Databases: Pradigm, Architecture, Survey and Issues.* Technical Report 15. Univ. of Karlsruhe, Dept. of Computer Science, Dec. 1988.

[LR94] M. Löhr and T. C. Rakow. *Audio Support for an Object-Oriented Database Management System.* Technical Report (Arbeitspapiere der GMD) 890. GMD St. Augustin, Dec. 1994.

[LS93] P. Lougher and D. Shepherd. The Design of a Storage Server for Continous Media. *The Computer Journal*, 36(1):32–42, 1993.

[Mas91] Y. Masunaga. Design issues of omega: an object-oriented multimedia database management systems. *J. of Information Processing*, 14(1):60–74, 1991.

[MW91] K. Meyer-Wegener. *Multimedia Datenbanken.* Leitfäden der angewandten Informatik. Teubner Stuttgart, 1991.

[MW94] K. Meyer-Wegener. Database management for multimedia applications. In *J. L. Encarnação and J. D. Foley (Eds.): Multimedia.* Springer Berlin, 1994.

[Nic90] C. Nicolaou. An Architecture for Real-Time Multimedia Communication Systems. *IEEE J. Select. Areas Commun.*, 8(3):391–400, 1990.

[OM88] J. Orenstein and F. Manola. PROBE Spatial Data Modeling and Queryprocessing in an Image Database Application. *IEEE Trans. Software Eng.*, 14(5), 1988.

[R+94] T. Rakow et al. Development of a Multimedia Archiving Teleservice using the DFR Standard. In *Proceedings of the 2nd International Workshop* *on Advanced Teleservices and high Speed Communication Architectures, LNCS*, pages 401–412. Springer Verlag, 1994.

[RJ93] L. Rabiner and B. H. Juang. *Fundamentals of Speech Recognition.* Prentice-Hall, 1993.

[RLM+93] T. C. Rakow, M. Löhr, F. Moser, E. J. Neuhold, and K. Süllow. Einsatz von objektorientierten Datenbanksystemen für Multimedia-Anwendungen (in German). *it+ti 3/93*, 1993.

[RM93] T. Rakow and P. Muth. The V3 Video Server - Managing Analog and Digital Video Clips. In *Proc. SIGMOD '93*, pages 556–557, May 1993.

[RSSS90] J. Rückert, H. Schmutz, B. Schöner, and R. Steinmetz. A Distributed Multimedia Environment for Advanced CSCW Applications. In *Proc. IEEE Multimedia*, 1990.

[RV93] P. Venkat Rangan and Harrick M. Vin. Efficient Storage Techniques for Digital Continuous Media. *IEEE Transactions on Knowledge and Data Engineering*, 5(4):564–573, 1993.

[RVR92] P. Venkat Rangan, Harrick M. Vin, and S. Ramanathan. Designing an On-Demand Multimedia Service. In *IEEE Communications Magazine*, July 1992.

[SE93] R. Steinmetz and C. Engler. *Human Perception of Media Synchronization.* IBM European Networking Center, 1993.

[SG94] P. Schäuble and U. Glavitsch. Assessing the retrieval effectiveness of a speech retrieval system by simulating recognition errors. In *Proceedings of the ARPA Workshop on Human Language Technology (HLT'94)*, 1994.

[SK91] M. Stonebraker and G. Kemnitz. The postgres next-generation database management system. *CACM*, 34(10):78–92, October 1991.

[SM83] G. Salton and M.J. McGill. *Introduction to Modern Information Retrieval.* McGraw-Hill, 1983.

[SR86] M. Stonebreaker and L. Rowe. The Design of POSTGRES. In *Proc. ACM SIGMOD*, 1986.

[Ste93] R. Steinmetz. *Multimedia-Technologie: Einführung und Grundlagen.* Springer Berlin, 1993.

[TR93] V. Turau and T. C. Rakow. *A Schema Partition for Multimedia Database Management Systems.* Technical Report (Arbeitspapiere der GMD) 729. GMD St. Augustin, Feb. 1993.

[TR94] H. Thimm and T.C. Rakow. A dbms-based multimedia archiving teleservice incorporating mail. In W.Litwin and T.Risch, editors, *Proceedings of the First International Conference on Applications of Databases (ADB)*, pages 281–298, Vadstena, Sweden, 1994. Lecture Notes in Computer Science 819, Springer.

[TY84] H. Tamura and N. Yokoya. Image Database Systems: A Survey. *Pattern Recognition*, 17(1), 1984.

[VAB95] M. Volz, K. Aberer, and K. Böhm. *A Flexible Approach to Combine IR Semantics and Database Technology and Its Application to Structured Document Handling.* Technical Report (Arbeitspapiere der GMD). GMD St. Augustin, Jan. 1995.

[WB92] Lynn D. Wilcox and Marcia A. Bush. Training and search algorithms for an interactive wordspotting system. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, March 1992.

[WK87] D. Woelk and W. Kim. Multimedia information management in an object-oriented database system. In *Proceedings of the 13th VLDB Conference,* Brighton, 1987.

[WKL87] D. Woelk, W. Kim, and W. Luther. Multimedia applications and database requirements. In *Proc. IEEE Computer Society Symposium on Office Automation*, April 1987.

[WYY91] J. Wells, Q. Yang, and C. Yu. Placement of audio data on optical disks. In *Int. Conference on Multimedia Information Systems '91*, pages 123–134. McGraw-Hill, 1991.

[YSB+89] C. Yu, W. Sun, D. Bitton, et al. Efficient placement of audio data on optical disks for real-time applications. *Communications of the ACM*, 32(7):862–871, July 1989.