Evaluating Object DBMSs for Multimedia

Paul Pazandak
Object Services and Consulting

Jaideep Srivastava
University Of Minnesota

We describe functionality for determining an object database management system's suitability for developing multimedia applications. We discuss all levels of hardware and software support, as even the most ideal database software cannot operate independent of operating systems, networks, and hardware. A review of the multimedia support provided by current commercial and research object database management systems is also included.

In the past, general database management systems (DBMSs) typically managed simple data types such as strings and integers. Simple record structures were sufficient to represent the managed data. More recently, object database management systems (ODBMSs)—based on the object-oriented data model—have provided better capabilities to manage more complex data requiring user-defined data types such as engineering designs and software configuration management. An increasing trend is to use DBMSs for managing multimedia data, particularly as software, networks, and computers improve in their handling of audio and video data requirements. While multimedia data has been stored in DBMSs since the 1980s, ODBMSs have generally become the database management system of choice (versus relational database management systems), because these systems support multimedia data better.[1] However, flat file storage still remains the most prevalent method (reasons for this will be discussed later).

This article describes the functionality required of ODBMSs to support multimedia data management. While others[1,2] have also addressed support requirements for multimedia databases, we consider the requirements for more sophisticated multimedia applications. Jain et al.[3] provide an excellent discussion of requirements for visual information management systems, which also highlights the dramatic changes necessary to support multimedia data.

In the future we will see distinctions between ODBMSs that provide simple multimedia support for multimedia data repositories and next-generation ODBMSs that support sophisticated distributed multiuser interactive and collaborative multimedia environments—which we call multimedia database management systems (mulitmedia DBMSs). See the sidebar "Multimedia DBMS Design Issues" for further reading.

Supporting multimedia data will require changes to ODBMSs' software, as well as operating systems, computer hardware, and networks. Although we focus on software metrics in this article, we will also discuss hurdles faced by other system components in satisfying the demands placed on them by multimedia data.

Multimedia data types

To understand the requirements that ODBMSs must satisfy, we need to know the types of multimedia data that can be stored and managed. Here's a list of the most common multimedia (or media) data types:

- *Text.* Large amounts of structured text in the form of books, for example, contain parts, chapters, sections, subsections, and paragraphs.

- *Graphics.* Graphics include drawings and illustrations encoded using high-level descriptions like CGM, Pict, and PostScript. This data type can be stored in a structured way within a database. You can easily query the content that exists as metadata, such as lines, circles, and arcs (for example, "find all graphics that contain a circle"). Of course, it's harder to find objects—like a chair—that are composed of simple types (such as lines and circles).

- *Images.* Images include pictures and photographs with encoding defined by standard formats such as bitmap, JPEG, and MPEG. The storage representation of images is a direct translation of the image, pixel by pixel, so no concept of a line, arc, or circle exists. Some formats, such as JPEG and MPEG, then compress the representation to reduce the size of the resulting data. Since you can't describe images by basic components such as lines, finding objects or complex objects within an image is difficult.

- *Animation.* A temporal sequence of image or graphic data, an animation specifies the order in which graphics or images should be rendered.

1070-986X/97/$10.00 © 1997 IEEE

Multimedia DBMS Design Issues

A number of projects have focused on multimedia data issues over the past fifteen years. A few of these include:

C.Y.R. Chen et al., "Design of a Multimedia Object-Oriented DBMS," *Multimedia Systems*, Vol. 3, No. 5-6, 1995, pp. 217-27.

T.-C. Chen et al., "Client-Server Database Environment for Supporting Multimedia Applications," *Proc. Int'l Computer Software and Applications Conf.,* IEEE Computer Society, Los Alamitos, Calif., 1994, pp. 215-200.

G.D. Drapeau and H. Greenfield, "Maestro—A Distributed Multimedia Authoring Environment," *Proc. Summer 1991 Usenix Conf.,* Nashville, Tenn., 1991, pp. 315-328.

K. Fujikawa et al., "Multimedia Presentation System 'Harmony' with Temporal and Active Media," *Proc. Summer 1991 Usenix Conf.,* Nashville, Tenn., 1991, pp. 75-93.

M.T. Ozsu et al., "An Object-Oriented Multimedia Database System for a News-on-Demand Application," *Multimedia Systems*, Vol.3, No. 5-6, 1995, pp. 182-203.

T.C. Rakow and M. Lohr, "Audio Support for an Object-Oriented Database Management System," *Multimedia Systems J.*, Vol. 3, No. 5/6, Oct/Nov 1995, pp. 286-297.

G.N.M. Sudhakar, A. Karmouch, and N.D. Georganas, "Design and Performance Evaluation Considerations of a Multimedia Medical Database," *IEEE Trans. on Knowledge and Data Eng.*, Vol. 5, No. 5, Oct. 1993, pp. 888-894.

D.L. Woelk, W. Kim, and W. Luther, "An Object-Oriented Approach to Multimedia Databases," *ACM Sigmod Record 1986*, Vol. 15, No. 2, pp. 311-325.

The images or graphics are independently constructed and organized. Unlike simple image data, which can be retrieved and viewed for any length of time, animation has a temporal viewing constraint requiring each image or graphic to be displayed and subsequently replaced by the next image or graphic. The constraint may vary by animation (it may be two images per second or thirty images per second).

- *Video*. Video is a set of temporally sequenced photographic data. The data represents a recording of a real-life event produced by a device such as a digital video recorder. The data divides into units called frames. Each frame contains a single photographic image. In most cases, video records at 24 to 30 frames per second (fps). Temporal viewing constraints are generally dictated by the recorded frame rates for optimal viewing.

- *Structured audio*. Like animation, this data represents a sequence of independent components having a temporal requirement. Each component is represented by using a description, such as note, tone, and duration. The temporal listening constraints can vary and are generally defined at creation time, or they're inherent to the component descriptions (for example, eighth notes).

- *Audio*. Audio data is a set of sequenced data generated from an aural recording. The basic units of audio data are called samples. Audio data has a temporal listening constraint dictated by the recording device's sampling rate for optimal playback.

- *Composite types*. Composite multimedia data is created by combining basic multimedia data types and other composite multimedia data. Types can be physically mixed together to form a new type or logically mixed. A physical mix results in a new storage format, where data such as audio and video intermix. A logical mix defines a new data type while retaining individual data types and storage formats. For example, a new type AV or audio-video would be composed of two distinct parts. However, when played, the executing methods would have to deliver the data in a synchronized fashion, making it appear as though the data is a composition. Composite data may also contain additional control information describing how the information should be rendered at the client.

- *Presentations*. Presentations are complex composite objects that also describe orchestrations of multimedia data for the purpose of modifying and presenting data. Orchestrations may describe a simple temporal ordering, such as playing video v1, then video v2, and so on. Or,

Acronyms

Blobs	binary large objects
CBR	content-based retrieval
CBQ	content-based querying
CGM	Computer Graphics Metafile
DBMS	database management system
DCBR	dynamic content-based retrieval
DML	data manipulation language
GIF	Graphics Interchange Format
IEC	International Electronics Commission
ISO	International Organization for Standardization
JPEG	Joint Photographic Experts Group
MIME	Multipurpose Internet Mail Extension
MPEG	Moving Pictures Expert Group
ODBMS	object database management system
ODMG	object database management group
O-RDMBS	object relational database management system
Premo	Presentation Environments for Multimedia Objects
QBE	query by example
QoS	quality-of-service
SCBR	static content-based retrieval
SGML	Standard Generalized Markup Language
SQL	Structured Query Language
VRML	Virtual Reality Modeling Language

they may be much more complex, specifying how user, system, and application interaction will determine the resulting presentation.[4]

Types of applications

Several uses **exist** for an object database management system that manages multimedia data. The application's requirements (and the kind of multimedia data it will use) can determine which features an ODBMS must support. For some applications, current ODBMSs can be used with little or no modification. For others, no ODBMS exists today (or for that matter, operating systems or computer hardware) that provides the necessary features.

Here we explain some typical applications that might use an ODBMS to manage multimedia data.

Data repositories

Data repositories provide simple database management support such as security and data backup. Repositories do not need to understand the stored data formats because they don't operate on the data. Transaction support is possible, but updates require replacing the entire object. Since the objects are stored as Blobs—or binary large objects—they exist as single "simple" objects. Queries may be formulated involving metadata and other data in the repository, but generally not against the multimedia data. In addition, repositories do not understand any temporal constraints inherent to the data, such as video. The data is simply sent to the clients, which have applications to handle it appropriately. A few examples of data repository applications follow:

- *Pseudo repository*. A pseudo repository contains multimedia metadata, such as the names, lengths, encodings, descriptions, and keywords of videos. Surrogate values stored within the database describe the pathname to the multimedia objects, which are stored as simple files within a local or network-accessible file system. An ODBMS has limited control of the files, since the files reside outside of the repository.

- *Simple repository*. A repository can provide restricted access to data and also a central location from which data can be backed up. Some applications may want a central managed storage facility to store and retrieve multimedia data. A DBMS manages the data, which can be stored on local disks or on tertiary storage devices such as optical disks. Applications retrieve the multimedia data, use it locally, and return it to storage.

- *Electronic mail*. Electronic mail may include sending multimedia data. The mail system may use a repository to store the multimedia data, or the data may originate from a repository. In either case, the repository acts as a server, merely sending the mail to the client when requested. To read the mail, the client must have an application that understands the multimedia data's format (for example, MIME).

- *Engineering designs*. For security and perhaps configuration management purposes, engineering drawings and solid models may be stored within a repository. Any operations, such as modification, are performed on the client using software that understands the data.

- *Healthcare information systems*. For archival purposes, patient data such as X-rays and doctors' annotations may be stored within repositories.

Intelligent data management

Because an ODBMS can also understand the data it manages, it can query multimedia object *content* and not simply metadata. Here we discuss some examples of applications that use an ODBMS for intelligent data management.

Working environments. Traditional database management systems offer basic creation, update, and query capabilities to standard data types. Extensions to these systems may enable the same support for multimedia data types. Examples include

- *Multimedia editing*. Since ODBMSs understand data formats, users can request portions of videos to update. ODBMSs support data-specific editing operations such as cut, paste, and crop.

- *Engineering design workflow*. In an engineering application, complex drawings are designed. These drawings need to be validated by mechanisms within the system whenever changes occur. These changes are validated against other parts of the design and existing design constraints, and updates are sent to affected components and engineers. Design changes can also influence documentation, causing new diagrams to be generated and inserted into appropriate sections of design and product documents.

- *Intelligent healthcare networks*. These systems let doctors collaborate by including media-related patient data within their interactive communications. In addition, features such as intelligent data routing may be supported—media data may be analyzed when stored, then routed to the appropriate healthcare specialists.

Presentation environments. ODBMSs can also deliver multimedia data that have temporal constraints, such as audio and video. In these applications, the data is consumed as it is delivered, unlike electronic mail. ODBMSs are well aware of the time sensitivity of data delivery. Example presentation environments include

- *Simple multimedia viewing*. Users retrieve multimedia data of interest and tell the system that they wish to view the data. As the data is retrieved from storage, it's immediately delivered frame by frame to the user. The user may have an interface similar to a VCR and be able to stop the selected video, fast-forward, play it in reverse, or jump to random points within the video. The delivered data satisfies the temporal viewing constraints.

- *Complex multimedia presentations*. Users retrieve composite multimedia for viewing—delivered frame by frame (or sample by sample) by the ODBMS. Orchestration directions, stored as metadata within the composite multimedia, dictate the retrieval order of each component, whether in series or in parallel.

- *Interactive multimedia environments*. These environments enable sophisticated database interactions including real-time editing, analysis or annotating of video and audio, interactive multiuser collaborations and presentations that can be driven by user, application, and system interaction, and advanced query capabilities.[5]

So, the basic uses of a multimedia database can be summarized as follows:

- *Read*. Retrieve and view data and presentations.

- *Update*. Includes creating new multimedia data and modifying data.

- *Compose*. Create compositions and presentations using basic multimedia data.

- *Query*. Search a multimedia database. Either indirectly querying multimedia metadata, or directly querying the actual data itself.

- *Interaction*. Includes user and ODBMS interaction with multimedia data. The data is no longer static but made dynamic by support for user (and multiuser), applications, and system interaction. In addition, data can interact and affect other data (such as engineering designs). Users define behavior during composition creation using a tool or language,[5] which supports this capability.

However, the metrics you must use to select a multimedia database management system will depend on the application and its requirements. As we have seen from the few examples above, ODBMSs can be used in a number of ways. Certainly most existing ODBMSs can act as pseudo or simple multimedia data repositories.

ODBMS functionality requirements

So what are the ramifications of providing multimedia data support with an ODBMS? We need to understand the features required that support our intended use of the data (reading, updating, querying, composing, and interacting).

Here we present an overview of the characteristics of multimedia data. We then visit each of these in more detail, describing the features an ODBMS must provide in its role as a multimedia DBMS.

- *Data types.* The ODBMS may treat multimedia data as Blobs (binary large objects), as in simple data repositories that do not recognize or support multimedia data formats. Otherwise, the ODBMS may provide support for several multimedia data types. Within object-oriented software these data types exist as class definitions. Thus, several class definitions might support all of these types. For each class definition, associated methods are included to support operations on the specified data types.

- *Data size.* Multimedia data size can be substantial. Even compressed movie formats result in 4 to 5 Gbytes of data for a two-hour movie. This fact alone can substantially affect the design of hardware and software.

- *Viewing.* In general, viewing a multimedia object requires retrieving it, then rendering it on a screen (or playing it on a speaker). For data such as audio and video, this requires the proper bandwidth at several points to satisfy the temporal constraints. In many configurations, the constraints may not be met. Therefore, users can indicate a level of quality-of-service (QoS) that satisfies their current needs. However, the environment may or may not be able to meet this service level at a given time, requiring a user to try again later or accept a lower QoS.

- *Querying.* Querying is common in any DBMS. Multimedia data must be interpreted before it can be queried. This process demands sophisticated indexing schemes and image and audio analysis algorithms to generate content descriptions. Users may want to query for images that "Look like this" or involve specific actions such as running. Thus, querying requires mechanisms to generate indices, interfaces and languages to pose queries, and underlying components to optimize queries.

- *Throughput.* To meet the requirements for playing audio and video data, we must optimize the software and hardware to ensure that it can satisfy the temporal constraints. A system's throughput is the primary reason these constraints are not met.

- *Resource scheduling.* A user may request the delivery of multiple audio and video streams in parallel from disk. In addition, multiple users can simultaneously request different data from the same disk. Finally, multimedia playback and recording devices must also be scheduled without conflict.

- *Memory, bus, CPU.* To handle multimedia data, such as rotating high-quality images, a computer must have sufficient main memory to load the images. Buffer strategies and the data bus and processor speed can also significantly affect the system's throughput.

- *Special chip sets and cards.* Due to increasing throughput requirements for audio and video, manufacturers have created specialized chip sets and boards for data capture, presentation, conversion, compression and decompression, and multimedia operations such as crop and rotate. In addition, companies such as Sun Microsystems developed CPUs that contain specialized instruction sets for handling multimedia data. These solutions have increased system throughput while also providing added functionality. These hardware products perform the same operations faster than their software counterparts, but they also cost more and are inflexible.

- *Storage.* Due to data size, you can store only two to three movies on the largest hard disk drives in mass production, which are about 9 Gbytes. This means that repositories for multimedia data must include larger storage arrays, such as disk arrays and CD jukeboxes for tertiary storage. Due to throughput requirements for videos, secondary storage devices must be fast enough to handle multiple requests. Therefore, parallel disk systems may be required.

- *Networks.* Finally, added concerns about throughput and reliability exist if data is delivered across a network. Current widely used protocols and hardware do not suffice for transporting high-quality video. Users will generally tolerate a lower quality picture over speed degradation (such as jitter).[6]

Supporting Next-Generation Multimedia DBMS Applications

As described above, the implications related to introducing and using multimedia are far-reaching, going well beyond software redesign. It's difficult to isolate those requirements necessary for an ODBMS to satisfy the demands of advanced multimedia applications. Operating systems, networks, and computer hardware must also evolve. In fact, newer versions of these ODBMSs may be built upon or integrated into optimized multimedia-aware operating systems, running on specialized hardware.

As previously stated, the features an ODBMS must provide vary from application to application. Therefore, we anticipate that each ODBMS product could be placed somewhere along axes describing its degree of functionality with respect to the multimedia features it provides. It's unclear at which point an ODBMS should be considered a multimedia DBMS, but in general it will depend on the application at hand.

In the following sections we focus on the functionality that a multimedia DBMS offers to meet the requirements of next-generation multimedia applications. While current ODBMSs provide some basic support for multimedia data, the next generation of applications will demand a significant evolution in functionality.

Data types

Multimedia DBMSs provide type support for common multimedia data types. Unlike simple repositories, multimedia DBMSs must understand the data they manage. They must include class definitions for several standard static multimedia data formats, such as JPEG, GIF, and MPEG, for noncontinuous data. Each class must have associated methods that can operate on the data. Internally, a multimedia DBMS may support a single (perhaps proprietary) data format for each data type, requiring data not in that format to be converted on import. Multimedia DBMSs must also provide direct class support for temporal (or "continuous") data types, including audio and video.

Eventually, a multimedia DBMS will support multimedia data types (particularly temporal data) as basic data types. Operating system support for multimedia data types will help to facilitate this. This low-level support for multimedia data types will enable ODBMSs to provide better optimized handling of them.

In most cases, current ODBMSs don't provide much support for multimedia data types. Most available support is for noncontinuous types with a limited number of operations. It will be some time before extensive support will be available for continuous data types.

Data size

Current ODBMSs generally do not support large amounts of video data. Database sizes may be limited (a constraint enforced by the file system used), or the ODBMS' design might restrict it from handling such large data files. This may be acceptable if the amount of data will be limited, or if the ODBMS acts as a simple data repository storing only metadata and filenames and not actual multimedia data. Otherwise, a multimedia DBMS should be capable of storing and managing several gigabytes for small multimedia objects such as images, and several hundred terabytes or more if the database must hold significant amounts of video or animation data. Supporting tertiary storage, such as CD jukeboxes, will be necessary to manage these large databases.

Data location

Most current ODBMS vendors recommend storing large multimedia files outside the ODBMS in the local file system. By storing the data in the file system, it prohibits the ODBMS from providing basic features such as fine-grain locking, transaction support, or recovery. (For example, surrogates in the ODBMS are locked, not the actual multimedia files stored in the file system.)

However, separating multimedia metadata from the multimedia data itself should be by design. Specifically, multimedia DBMSs need to provide optimized storage managers that deliver continuous data in real time—these managers also provide all of the features of a basic storage manager, such as fine-grain locking. While optimized managers

Modeling Layers

Temporal and spatial data models will play an important role in Multimedia DBMSs. These articles describe layered data models.

K. Aberer and W. Klas, "Supporting Temporal Multimedia Operations in Object-Oriented Database Systems," *IEEE Int'l Conf. on Multimedia Computer Systems*, IEEE Press, Piscataway, N.J., May 1994.

A. Gupta, T.E. Weymouth, and R. Jain, "An Extended Object-Oriented Data Model For Large Image Databases," *Second Symp. SSD 1991—Lecture Notes in Computer Science*, Vol. 525, Springer, Berlin, 1991, pp. 45-61.

S. Marcus, *Multimedia Database Systems*, tech. report, Mathematical Sciences Institute, Cornell University, http://www.cs.umd.edu/projects/hermes/publications/postscripts/mm1.ps.

G. Schloss and M. Wynblatt, "Building Temporal Structures in a Layered Multimedia Data Model," *Multimedia 94: 2nd Annual Conf. on Multimedia*, ACM Press, New York, 1994.

only store the actual data having strict temporal access constraints, the data's metadata (indices and keywords) can be stored in a standard manager.

Data model

The richness of the data model plays a key role in its usability. Although multimedia data types must be supported, they only provide the foundation to build additional features. Here we describe the support that a multimedia DBMS' data model must provide.

Multimedia frameworks. In most cases, multimedia data types are implemented as classes, having attributes and methods to support each data type. They may be part of a large class hierarchy or framework, composed of similar data types (for instance, all sound-based data types or all graphic data types). A framework goes beyond a simple type definition to give more complete support for the data types, including several methods and additional supporting classes (to import, view, and export). These frameworks provide capabilities comparable to stand-alone commercial multimedia editing applications. In fact, ODBMS vendors may license and plug in these commercial applications rather than invest in internal design, development, and maintenance of specialized code. This happened with ODBMS vendors that licensed the Virage[7] visual retrieval engine, which provides advanced image querying support.

Support for multimedia relationships. While type frameworks focus on type-specific support, several kinds of relationships need to be expressed between multimedia data for composition and presentation. These relationship semantics do not exist in current ODBMSs. However, the relationships can be expressed by augmenting standard relationships with additional methods and classes (in addition to providing low-level support for these relationships).

You can view multimedia data in a random fashion or as organized multimedia presentations. Unlike consumption of a single image, presentations have spatial and/or temporal dimensions. To support them, you must define relationships between parts of the presentation to provide spatial and temporal structure. Spatial structures support the definition of books and papers—the layout of information that has no temporal constraints. Temporal structures let temporal dynamics be specified, that is, when data should be played. In contrast, simple composition available in current ODBMSs describes associations without temporal or spatial structure.

These relationships define additional modeling layers on basic multimedia data. The layers required for any multimedia data model vary depending on use. For image analysis, models such as those used by Gupta et al. may be practical, while presentations require temporal and spatial layers. These layers provide and organize information so that it can be intelligently consumed (see the sidebar "Modeling Layers" for further reading).

Spatial relationships organize the data's visual layout on a virtual page or medium. The virtual medium may exist across multiple machines. Within a spatial presentation, users can move around inside the boundaries defined by the virtual medium and move and restructure the data.

Spatial structures may include 3D or virtual environments. Spatial constraints control **3D** object movement and interobject spatial relationships. Special tools can define spatial relationships using graphical user interfaces. Instead of providing semantically rich spatial relationships, some systems support spatial grammars,[8] which are closer to scripting languages. However, since these spatial relationships are not directly represented within the database (but only within a script), they cannot be readily queried.

Temporal structures dictate the temporal layout, orchestrating the data's presentation. A simple example might be, "Play video v1 and audio a1, and, when finished, play music m1 until slide show s1 is done." Basic temporal structures produce serial and parallel (hierarchical model[9]) presentations of data. You can also define presentations by associating a presentation time and duration (timeline model[10]) with each multimedia object, eliminating the requirement for temporal structures. However, the model is severely limited, since it can only define static presentations. Other approaches for temporal specification include scripting languages, specification languages, and extended programming languages.

While specification languages may be mapped to temporal relationships within a multimedia DBMS, it is not so easy to do the same with scripts and programs. Scripts and programs control the presentation of multimedia data, but without a representable temporal structure they cannot easily be queried, reused, or supported by the underlying components of the multimedia DBMS (such as the storage manager).

More advanced temporal models build on the event-driven model, enabling user, application, and system events (interaction) to affect the runtime presentation (for further reading, see the sidebar "Advanced Temporal Models"). In fact, with complex user interaction supported, some multimedia DBMSs can become the infrastructure for a distributed interactive multimedia environment[5] supporting multiple users (user-user interaction), guided instruction, collaboration, and interactive virtual worlds.

In addition to temporal relationships and temporal and spatial constraints, we need control structures to describe the control of the presentation such as fine-grain synchronization constraints, delivery constraints, and presentation constraints. Fine-grain synchronization controls the temporal presentation of two or more multimedia objects, such as maintaining lip synchronization between a movie and a soundtrack. Delivery constraints specify QoS parameters and alternative presentations depending on resource availability. Presentation controls enable a high level of control over a presentation, like applying special effects.

Device hierarchies. Capturing and presenting multimedia data involve hardware devices such as cameras, microphones, computer cards, speakers, monitors, and other equipment. By modeling these devices logically as classes, the underlying hardware details can be abstracted away from the user. This lets the entire process (capture, storage, and presentation) be represented within the data model. By providing this support, it's easy for the application designers (and multimedia DBMS vendors) to extend the device hierarchy to add additional devices as required. Since the devices exist as abstractions, some devices may actually be implemented completely in software. Finally, if devices are simply objects within the database, they can be used without regard to location (location transparency).

Advanced Temporal Models

While layered data models provide part of the solution, event-driven temporal synchronization models provide support for defining interaction and presentation semantics.

M.C. Buchanan and P.T. Zellweger, "Automatic Temporal Layout Mechanisms," *Proc. ACM Multimedia 1993,* ACM Press, New York, 1993, pp. 341-350.

M. Vazirgiannis and M. Hatzopoulos, "Integrated Multimedia Object and Application Modeling Based on Events and Scenarios," *Int'l Workshop on Multimedia Database Mgmt Systems (IW-MMDBMS 95)*, IEEE Computer Society Press, Los Alamitos, Calif., 1995, pp. 48-55.

P. Pazandak, *Multimedia Language Constructs and Execution Environments for Next-Generation Interactive Applications*, PhD thesis, University of Minnesota, Dept. of Computer Science, Minneapolis, Minn., 1996.

Use implications

Beyond data model and device support, multimedia DBMSs must provide features related to user interaction, data manipulation, and data query.

User interaction. Multimedia applications demand user interaction support, such as sophisticated graphical user interfaces. Because multimedia DMBSs require spatial and temporal rendering, the interface design must include features to control and devices to render the presentation. This may include simple VCR-type control panels enabling the user to play, fast-forward, pause or rewind a presentation, as well as more advanced controls including interactive data filtering, querying, and visual perspective controls. In addition, they must support multiple viewing definitions of the data. The user must also control the QoS parameters to adjust for resource limitations, cost of delivery, and personal visual and aural preferences.

User interaction must also support user events, whereby the system detects user interaction with the data, like the event manager in Damsel,[4] or the interaction manager in Vodak.[11] Some ODBMSs support basic event handling, but the system's response delay to events may not be acceptable to some users. To support advanced multiuser collaboration and video interaction, multimedia DBMSs provide soft, real-time multithreaded functionality. As the user interacts with

Optimizing Query Techniques

A number of articles from a recent international Multimedia DBMS workshop address many issues related to querying multimedia data. Quite a bit of research exists in this area, and importantly, companies such as Virage (www.virage.com) are helping to commercialize this research for use in current DBMS products.

S.T. Campbell and S.M. Chung, "The Role of Database Systems in the Management of Multimedia Database Management Systems," *Int'l Workshop on Multimedia Database Mgmt. Systems (IW-MMDBMS 95)*, IEEE Computer Society Press, Los Alamitos, Calif., 1995, pp. 4-11.

M. Ghandi, E. Robertson, and D.V.Gucht, "Modeling and Querying Primitives for Digital Media," *Int'l Workshop on Multimedia Database Mgmt. Systems (IW-MMDBMS 95)*, IEEE Computer Society Press, Los Alamitos, Calif., 1995, pp. 82-89.

S. Hibino and E.A. Rundensteiner, "A Visual Query Language for Identifying Temporal Trends in Video Data," *Int'l Workshop on Multimedia Database Mgmt. Systems (IW-MMDBMS 95)*, IEEE Computer Society Press, Los Alamitos, Calif., 1995, pp. 74-81.

N. Hirzalla and A. Karmouch, "A Multimedia Query Specification Language," *Int'l Workshop on Multimedia Database Mgmt. Systems (IW-MMDBMS 95)*, IEEE Computer Society Press, Los Alamitos, Calif., 1995.

several different windows and data types, the systems must support different threads of control[2] within each user process.

Data manipulation. One purpose of a DBMS is to provide support for data manipulation. This task proves trivial for simple data types, but not for multimedia data. To modify data within multimedia DBMSs, they integrate advanced editing environments with tool and language support. Although editing tools provide interactive editing, multimedia databases require multimedia data manipulation languages (DMLs) or constructs. In the Object Database Management Group (ODMG-93), the DML is the programming language. However, standard programming languages do not support temporal data.

Multimedia DMLs support data flow, indicating sources, sinks, and operations in between to analyze and modify the data. However, simple cut or crop operations do not suffice; these operations must adhere to the real-time delivery constraints. For example, when watching a video, a user may decide to crop and rotate it. To maintain the temporal constraints, these operations must provide predictable and worst-case information about the operation overhead. This lets the system determine if QoS guarantees can be met when the operation is inserted within the data stream between the source and the sink (the viewer).

Once modified, the data can be stored again in the database. It may replace the old version or exist as another version of the data. If users constantly create new versions of data, the database will grow rapidly. To reduce the need to store new versions of data, a multimedia DBMS stores the operations to generate the new data—called "derived data"—rather than storing the new data itself.

Querying. You can retrieve stored data by initiating queries. Queries contain predicates that must be satisfied by any data retrieved. The predicates usually involve partial or exact matches, such as "Find all employees whose last-name is Doe," and value ranges, such as "Find all employees whose salary is between $2,000 and $2,500." But how do you query multimedia data?

Like data manipulation, multimedia DBMSs must support a multimedia query language. However, at this point such issues remain unsolved. When multimedia DBMSs start to provide multimedia query languages, techniques must also be developed to optimize queries in these languages (for more information, see the sidebar "Optimizing Query Techniques").

Manual keyword indexing remains the most straightforward method to query multimedia data. Storing multimedia data in a database generates descriptive keywords associated with the data (these keywords are metadata, since they are data about multimedia data). It's important to use a standardized keyword dictionary, hierarchical taxonomy, or thesaurus so that all classified data uses the same terminology. When a user wants to find a white house with a bay window in front, the DBMS examines keywords of all house images stored in it. The images themselves are not queried.

The keyword approach does have problems. First, keyword classification remains subjective, since it is performed by a human. Second, exceptions will always exist, and some data may be incorrectly classified. Lastly, keywording is usually limited to a well-defined abstraction of the data (for example, for each house image, a specific set of features is classified). This means that if the abstraction becomes altered, then all of the data must be reviewed again, adding new keywords as required. Even with a small database, this could be a formidable task.

However, keywording enables fast retrieval of data. Standard indexing approaches can be used, since the keywords (strings) are a data type supported by every DBMS. For specialized applications, such as real estate image databases, this is probably all that is required. Unfortunately, the classification of videos may be overwhelming. Considering that a video is a sequence of (related) images, classification could require indexing each scene—and there may be thousands within one video.

A second approach that multimedia DBMSs must support is content-based retrieval/querying (CBR/CBQ). CBR bases queries on the content of multimedia data. Data analysis must be performed, but it can be done by audio and image analysis algorithms. Therefore, as better algorithms become available, the data may have to be reprocessed. Data analysis generally takes place when the data enters the database. The results of the analysis may be keywords or multidimensional indexing structures describing the data (see the sidebar "Describing the Data Through Analysis" for further reading"). Queries against this data then take place on the derived data, whose goal is to minimize the data abstraction by describing the data as completely as possible. The derived data could include attributes such as lines, shapes, colors, and textures from which objects could be determined at some later time. As these algorithms become more sophisticated, they will help reduce—but not eliminate—human-generated indexing.

Note that querying issues dealing with the various multimedia data types are nontrivial, so it will be some time before multimedia DBMSs provide all of the sophisticated query support we describe here. Analyzing video and detecting actions is currently far from reality. However, as the importance of multimedia databases grows, so will the demand to intelligently query them. To distinguish CBR on temporal data such as video and audio, from that on images and graphics, we call CBR on images "static content-based retrieval," or SCBR; and we call CBR on video "dynamic content-based retrieval," or DCBR.

With both types of CBR, specifying precise queries (exact matching) generally is not practical (or even possible in some cases). Therefore, multimedia DBMS query languages must be augmented with fuzzy predicates such as "like" to find approximate matches (for more information, see the sidebar "Fuzzy Predicate Research").

In addition, new query interfaces must support CBR. These interfaces will let the user specify the interesting attributes by providing examples to match—such as a drawing, photograph, action, or sound. The query manager can then use that example to find other similar examples in the database. This approach is called query by example (QBE). Some projects have already implemented static QBE in research systems for image retrieval. In static QBE, the users draw an example of the image they want to retrieve, using shape, color, or texture. In dynamic QBE, a user would illustrate an action to describe an example of the video or audio sequences that should be returned. Current technology does not support dynamic QBE.

Finally, as many data formats exist for each data type, either analysis code must be provided for each format or all data of a given type (such as image) must translate into one standard format. A problem with data formats, however, is that formats for analysis may require more space than compressed formats. For example, MPEG enables a 100:1 compression on video, but no algorithms

Describing the Data Through Analysis

In conjunction with automated indexing of multimedia content (which is critical to the long-term success of multimedia DBMSs), associated indexing structures are required to store this information for efficient content-based retrieval. Some of these issues are addressed in the following articles:

A. Gupta, T.E. Weymouth, and R. Jain, "An Extended Object-Oriented Data Model For Large Image Databases," *Second Symp. SSD 1991—Lecture Notes in Computer Science*, Vol. 525, Springer, Berlin, 1991, pp. 45-61.

Y. Niu, M.T. Ozsu, and X. Li, *A Study of Indexing Techniques for Multimedia Database Systems*, Tech. Report TR 95-19, Department of Computing, University of Alberta, 1995.

Q. Yang, A. Vellaikal, and S. Dao, "MB+Tree: A New Index Structure for Multimedia Databases," *Int'l Workshop on Multimedia Database Mgmt. Systems (IW-MMDBMS 95)*, IEEE Computer Society Press, Los Alamitos, Calif., 1995.

Fuzzy Predicate Research

A key feature required of all multimedia DBMSs will be a capable query interface and search engine. Precise queries will be replaced by static and dynamic QBE. Query languages and interfaces to support static QBE are addressed in the following papers:

A. Cardenas et al., "Knowledge-Based Object-Oriented PICQUERY plus Language," *IEEE Trans. on Knowledge and Data Eng.*, Vol 5, No. 4, Aug. 1993, pp. 644-657.

L.H. Chom, "Developing a Text and Image-Based Database Production System and Search Engine," *16th National Online Meeting Proc.*, Learned Information, New York, May 1995, pp. 53-55.

J.K. Wu et. al., "CORE: A Content-Based Retrieval Engine for Multimedia Information Systems," *Multimedia Sys.*, Vol. 1, No. 3, 1995, pp. 25-41.

Real-Time Delivery

We distinguish between *best effort* and *deterministic* real-time delivery, since a number of vendors claim they provide real-time delivery. Best effort (which some DBMS products support) means that the system cannot ensure quality of service constraints, while deterministic delivery provides guaranteed quality of service (which no DBMS product supports). Deterministic delivery requires operating system-level admission control policies to ensure that new processes do not overburden a system's resources. This ensures that current processes will be provided the level of service required to meet the QoS constraints specified by the application or user.

currently exist that can properly analyze the data in this format. In addition, many compression algorithms are not lossless, so the repeated compression and decompression of media data using them will degrade data quality.

System infrastructure issues

Given the nature of multimedia data and its usage requirements, the multimedia DBMS' system infrastructure must address new issues. Some of these issues follow.

Performance

Since multimedia data is generally large, throughput for most applications remains critical. For static data repositories it might not be important because the user may tolerate some delays. In addition, if data activity doesn't involve a user (for example, offline data analysis), then the responsiveness may not be important. However, when human consumption of temporal data is involved, throughput performance becomes critical. For single-user applications, the effort needed to support the throughput requirements will decrease. But, in most cases, applications will support multiple users accessing multiple audio and video objects simultaneously.

For example, temporal data could be delivered prior to viewing, requiring users to wait for the data to arrive—data repositories use this approach. However, this means that users may have to wait a few hours or more to start watching a video. In more critical application areas, such as defense, users will want to start watching and listening to the videos and high-fidelity audio immediately, and without jitter. Thus, multimedia DBMSs need to regulate delivering the data, frame by frame, to the user.

Unlike simple data types, and even long text that may be several megabytes in size, for real-time consumption, temporal data requires regulated data delivery (see the sidebar "Real-Time Delivery"). This forces the multimedia DBMS to manage access to the disk, negotiate delivery schedules, and set user-process priority levels to ensure QoS requests are met. To understand the delivery constraints, the storage manager—which retrieves the data—must integrate into the multimedia frameworks, which understand the high-level requirements of each media type and the QoS parameters set by the application or the user. Problems that occur during delivery will probably not be handled by the storage manager, but rather by the software or methods that control delivery of the information. Integrated solutions focus on multilayered specification and distributed resource management for continuous data.

In an ODBMS, data are generally placed on disk pages wherever space is available. Some systems also support data clustering, so the storage manager co-locates data on the same page to minimize page faults during retrieval. Due to the temporal constraints of continuous data, multimedia DBMSs will need to provide more rigorous placement algorithms.[12] Since each multimedia data object consumes many pages, these algorithms must focus on placing the data on disk (or disk arrays) to optimize retrieval. Placement may consider the expected retrieval pattern. However, further complications arise when multiple streams must be synchronized. Other retrieval algorithms may replicate data to satisfy multiuser demand.[13] These important issues directly affect the system's throughput.

To complicate things, compression algorithms such as MPEG may generate compressed video frames of unequal size. This hinders placement and retrieval of algorithms, since the compressed frame sizes cannot be predicted. Therefore, when retrieving the data, the required disk, buffer, and network resources will vary dramatically. Compressed streams of this kind are called variable bit-rate streams. Other kinds of data, including audio, compress predictably, and thus produce constant bit-rate streams that are easier to handle.

We must also consider the execution environment when defining throughput support requirements. If the multimedia DBMS server and the user work directly on the same machine, it reduces the complexity of data delivery. Otherwise, if the user (client) works on a different machine than the server, the network must be considered. We explore both configurations below.

Single machine. Single-user systems alleviate many of the problems that multiuser systems must address. With the user sitting at the server, no network concerns exist with respect to data retrieval if all data is locally stored. However, if you must retrieve data from devices across a network, then the same problems occur as those in the client-server configuration. Considering the size of future multimedia databases, it's very likely that tertiary storage will be used. In a single-user configuration, multimedia DBMSs let the user application access the multimedia data in shared memory to minimize internal access times.

Client-server. In a client-server configuration, data must be delivered to the client machine. However, current network speeds available through the Internet and local Ethernet, for example, do not suffice for high-quality video and fidelity audio data. In addition, the protocols do not support the time-constrained delivery of data, nor do they support QoS parameters.

In addition, high-speed networks that understand QoS parameters will be necessary to meet bandwidth requirements for video. For applications involving images, graphics, and audio, this won't be as serious a problem. Clients may connect to multimedia DBMS servers using high-speed communications to ensure that QoS requests can be met. Multimedia DBMS products will provide some level of support for audio and video streams—ranging from support of two simultaneous streams (perhaps acceptable for a single user) to more than 1,000 simultaneous streams (suitable for many users).

A client-server configuration must also accommodate device distribution, device scheduling, and data delivery. Because devices such as capture boards, cameras, VCRs, and special monitors available on the server or the network will be shared by multiple users, the multimedia DBMS must support device transparency and device sharing.

Operating systems

Due to the time-critical nature of temporal multimedia, multimedia DBMSs providing real-time access to video data will significantly benefit from multimedia-specific operating systems. Until these become available, soft real-time operating systems will likely be used. It's important to understand that much of the software functionality described in this article only provides part of the solution. In that regard, real-time operating systems can only be useful when other components, such as the multimedia DBMS software and network protocols, can be integrated to satisfy the various constraints associated with multimedia capture and presentation.

As computers and storage devices become faster, the number of users accessing a database will increase. In a single-user environment, with a relatively fast DBMS, real-time operating systems may not be required. For multiuser systems, multimedia DBMSs will require accurate control of data delivery. Plus, they must prioritize threads of execution, to ensure that QoS levels are maintained. To aid in meeting these delivery constraints, new buffer management strategies will be necessary within operating systems. In addition, to satisfy QoS constraints, systems will have to define admission control policies based on resource availability and other factors, such as priorities.

Related issues

Several interesting issues pertaining to multimedia DBMSs exist. For example, what distribution capabilities will be supported? Particularly with the evolution toward widely distributed database systems, what architectures will be used? Of course, within distributed systems, multiple languages will likely interface with each database. So, unlike several current ODBMSs, multimedia DBMSs should support language-independent storage of multimedia data (see Xerox PARC's Inter-Language Unification project at ftp.parc.xerox.com/pub/ilu/ilu.html, or the Object Management Group CORBA standard at www.omg.org).

What kinds and levels of authorization will be supported within each multimedia DBMS? Portions of audio or video may be classified, or an area within a video or image may be classified. When the video is played back, the restricted portions are hidden (cropped or masked before sending the frames to the client). It's also possible that videos may be viewed separately, but not simultaneously. Or, that parts of conversations must be blocked out. History-based authorization that restricts future accesses based on previous ones may also be used. Multimedia DBMSs will eventually provide more sophisticated authorization schemes.

There are also issues related to system extensibility. As software and hardware evolve rapidly in the multimedia domain, systems that offer some level of plug and play capabilities will enable a faster evolution and customizability of their software.

Internet access

Since more people continue to access the Internet, it makes an attractive venue for offering

Table 1. Status of ODBMSs (and O-RDBMSs), multimedia DBMS products, and research prototypes.

Feature Category	ADB Matisse	Basesoft EasyDB	CINCOM	Gemstone Gemstone	GMD AMOS	IBEX Itasca	Informix Illustra	Fujitsu/CAI Jasmine	Kala Kala
Commercial Product?	YES	YES	YES-UniSQL	NR	-	YES	YES	NR	YES
BLOB Support?	YES	YES	YES	-	YES	YES	YES	-	YES
Maximum Object Size	4GB	2GB	h/w	-	h/w	h/w	h/w	-	h/w
Storage Location:									
File System	YES	YES	YES	-	-	YES	YES	-	YES
DBMS	YES	YES	YES	-	YES	YES	YES	-	YES
Maximum Size of DB	-	2^122	h/w dep.	-	h/w dep.	h/w dep.	100's gb	-	h/w dep.
MM Class Support:									
Type Support	-	-	AI	-	IVACP	AI	IV	-	-
User-Defined	YES	YES	YES	-	YES	YES	YES	-	YES
Low-level Types	-	POSIX	-	-	YES	-	-	-	-
Other	-	-	-	-	-	-	-	-	-
Keyword Indexing	-	-	-	-	YES	-	Verity	-	-
Content Indexing	Virage	-	-	-	-	-	Virage	-	-
Temporal Model	-	-	-	-	Hier.	-	-	-	-
Language Support	-	-	-	-	YES	-	-	-	-
Spatial Relations	-	-	-	-	-	-	-	-	-
Constraints	semantic	-	-	-	temporal	-	sql rules	-	-
Device Hierarchies	-	-	-	-	-	-	-	-	-
Other Factors	-	-	-	-	YES	-	-	-	-
Frameworks	-	-	-	-	-	-	-	-	-
Querying Support									
Query MM Data	YES	-	YES	-	YES	YES	YES	-	YES
Query Language	sql,tools	o-sql	sql+mthds	-	OQL	Itasca	sql	-	-
Fuzzy Queries	YES	-	-	-	-	-	YES	-	-
Content-Based	-	-	-	-	-	-	-	-	-
- Static	Virage	-	-	-	-	-	Virage	-	-
- Dynamic	-	-	-	-	-	-	-	-	-
Other	-	-	-	-	-	-	-	-	-
Multimedia Tools									
Tools	-	-	OLE	-	A/V	-	-	-	-
Off-line / RT	-	-	-	-	Off-line	-	-	-	-
GUI	-	-	-	-	-	-	-	-	3rd party
Require filesystem	-	-	-	-	NO	-	-	-	-
Vendor Tools	-	-	-	-	-	-	-	-	Acrobat
ODBMS Description									
Languages	6	4	4+ODBC	-	2	5	4	-	7
Language Independ.	YES	YES	YES	-	NO	YES	YES	-	-
Versioning	YES	YES	-	-	-	YES	ts	-	YES
Change Notification	YES	-	-	-	-	YES	rules	-	YES
Dynamic Schema	YES	YES	YES	-	-	YES	?	-	YES
Data Delivery									
Best Effort Delivery	YES	YES	-	-	YES	YES	-	-	OS
- Constraints?	client	-	-	-	Preload & Buf., QoS	-	-	-	dynamic buffering
Real-time Delivery	-	-	-	-	-	-	-	-	-
Constraints?	-	-	-	-	-	-	-	-	-
Stream Synchro.	-	-	-	-	YES	-	-	-	-
Realized throughput	-	-	-	-	-	-	-	-	-
Platform Support									
Hardware	1,2,4,5	1,2	1,2,5	-	1	1	1,2	-	1,2,3
Real-time OS	-	encore,+	YES	-	-	-	-	-	ANSI C
Storage Management									
Specialized Manager	-	YES	-	-	YES	YES	-	-	YES
Extensible	YES	-	YES(?)	-	-	-	YES	-	n/a
Direct Access	YES	-	-	-	-	-	YES	-	-
Other Extensibility	YES	-	-	-	DML/DDL	kernel	-	-	all
Information via Non-disclosure Agreement?	YES	-	YES	-	-	-	YES	-	-

Hardware Support: 1-UNIX, 2-NT, 3-Mac, 4-Win, 5-VMS 6-RS/6000

Type Support: (I)mage,(A)udio,(G)raphics,(V)ideo,a(N)imation,(C)omposite,(P)resentation

h/w dep. - Hardware Dependent
NR - No Response - We did not receive a response from this vendor.

Disclaimer: No effort has been made to substantiate the information provided by these vendors.

access to databases. Many databases and repositories are already accessible via the World Wide Web. But, until the Internet can offer sustained transfer rates suitable for multimedia video, users will not have access to high-quality video and audio data in (deterministic) real time. Of course, a user can download an entire video or audio file and then view it, or receive best-effort delivery of video and audio with a fairly low QoS.

Standards impact

Currently, the object database vendors' standards group (ODMG, see http://www.odmg.org) has not addressed issues related to support for

Feature Category	Mediaway MediaDB	Neologic	ODI Object Store	Ontos Ontos	O2 O2	Persistence / Oracle	Poet Poet	Sybase Sybase	UAlberta	Versant Versant
Commercial Product?	YES	YES	YES	YES	YES	NR	YES	NR	-	NR
BLOB Support?	YES	YES	YES	LgTxt	YES	-	YES	-	YES	-
Maximum Object Size	4GB	4GB	4GB	h/w	2GB	-	2GB	-	2GB	-
Storage Location:										
File System	YES	YES	YES	YES	YES	-	YES	-	MM file sys.	-
DBMS	YES	YES	YES	YES	YES	-	YES	-	-	-
Maximum Size of DB	256TB+	4GB	unknown	h/w dep.	2^104	-	2GB	-	unknown	-
MM Class Support:										
Type Support	"MOST"	-	IVAT...	-	IGAVA	-	OLE BLOB	-	IVAP	-
User-Defined	YES	YES	YES	YES	YES	-	YES	-	YES	-
Low-level Types	-	-	-	-	-	-	-	-	-	-
Other	-	YES	-	-	-	-	-	-	-	-
Keyword Indexing	hash/Btree	-	-	-	Verity	-	-	-	YES	-
Content Indexing	-	-	Virage	-	-	-	-	-	-	-
Temporal Model	-	-	-	-	-	-	-	-	HyTime	-
Language Support	-	-	-	-	-	-	-	-	-	-
Spatial Relations	-	-	-	-	-	-	-	-	HyTime	-
Constraints	-	-	-	-	-	-	-	-	-	-
Device Hierarchies	-	-	-	-	-	-	-	-	-	-
Other Factors	-	NeoMedia	-	-	spatial indexing	-	-	-	-	-
Frameworks	-	-	-	-	-	-	-	-	-	-
Querying Support										
Query MM Data	YES	YES	YES	-	YES	-	YES	-	YES	-
Query Language	sql	Neo	sql	-	oql	-	oql	-	ODI API	-
Fuzzy Queries	-	-	YES	-	-	-	-	-	-	-
Content-Based	-	-	-	-	-	-	-	-	-	-
- Static	-	-	Virage	-	-	-	-	-	-	-
- Dynamic	-	-	-	-	-	-	-	-	-	-
Other	-	-	-	-	-	-	-	-	Doc. Structs	-
Multimedia Tools										
Tools	-	-	-	-	Look/Graph	-	-	-	code gen.	-
Off-line / RT	-	-	-	-	-	-	-	-	-	-
GUI	-	-	-	-	-	-	-	-	-	-
Require filesystem	-	-	-	-	-	-	-	-	-	-
Vendor Tools	-	HTML,C++	-	-	-	-	-	-	SGML	-
ODBMS Description										
Languages	3	1	4	1	5	-	3	-	1	-
Language Independ.	YES	-	-	YES	YES	-	YES	-	-	-
Versioning	YES	YES	YES	YES	YES	-	YES	-	-	-
Change Notification	YES	YES	YES	-	YES	-	YES	-	-	-
Dynamic Schema	YES	YES	YES-Smalltalk	YES	YES	-	YES	-	-	-
Data Delivery										
Best Effort Delivery	-	-	-	-	YES	-	YES	-	-	-
- Constraints?	buffering	-	-	-	-	-	methods	-	QoS	-
Real-time Delivery	-	-	-	-	-	-	-	-	-	-
Constraints?	-	-	-	-	-	-	-	-	-	-
Stream Synchro.	-	-	-	-	-	-	-	-	-	-
Realized throughput	-	-	-	-	-	-	-	-	-	-
Platform Support										
Hardware	1,2	1-4	5	1	1-3+	-	1-4	-	6	-
Real-time OS	-	YES	LynxOS	-	-	-	YES	-	-	-
Storage Management										
Specialized Manager	YES	-	YES	-	YES	-	-	-	-	-
Extensible	YES	YES	YES	YES	YES	-	-	-	-	-
Direct Access	YES	YES	YES	YES	YES	-	-	-	-	-
Other Extensibility	-	all	-	all	indexing	-	-	-	-	-
Information via Non-disclosure Agreement?	-	-	-	-	-	-	-	-	-	-

multimedia data. However, the frameworks and internal architectures required for multimedia DBMSs can still be supported while maintaining compatibility with the ODMG standards. Other standards efforts will also affect multimedia database systems. One such standard is ISO/IEC 14478—Presentation Environments for Multimedia Objects (Premo). It focuses on many issues related to multimedia data type support, synchronization, and programming.

Standards compliance is important for portability of data between products, such as the ISO SGML/HyTime multimedia interchange standard. As multimedia DBMSs become a reality, standards like Premo and SGML—as well as Internet standards, like the Virtual Reality Modeling Language (VRML)—will play an integral role.

Current efforts

Table 1 shows the current state of ODBMSs and

O-RDBMSs. Most of these systems don't have the necessary functionality to support multimedia, thereby limiting the ODBMSs to act as data repositories. Certainly, most systems offer basic class support to store images, while few have class support for audio and video. The class support these systems provide stores the data within the database as typed objects. However, the classes generally provide a set of very limited options. In addition, the systems tend toward simple class structures rather than complex multimedia frameworks. For more information on this survey, visit http://www.cs.umn.edu/~pazandak/MMDBMSMetrics/mmdbms-surveyTable.html.

Table 1 primarily shows only those features currently offered by the ODBMS products. We labeled the hardware as follows: 1. Unix; 2. Windows NT; 3. Macintosh; 4. Windows; 5. VMS; and 6. RS 6000. Hardware-dependent specific programming languages are not listed. In the multimedia class support feature category, we defined type support as A for audio, C for composite, G for graphics, I for image, N for animation, P for presentation, and V for video.

Most of these systems should be able to store image data, at least as Blobs. Since video data is so large, most systems in Table 1 cannot handle it, or handle it easily. If the systems could support extensible storage managers, then you could build and integrate dedicated, specialized continuous storage managers into the system. These managers could provide the throughput necessary to satisfy the temporal constraints (over reliable QoS-aware networks). While most vendors listed in Table 1 claimed they had extensible storage managers, their proposed implementations amounted to a pseudo-repository approach. Integrating a specialized storage manager, or a more enhanced type manager, for multimedia data would require additional work for the vendors, provided that a specialized storage manager already exists. The Presto Continuous Media File Server, one such storage manager, is currently being completed under an Air Force contract as a joint Honeywell Technology Center-University of Minnesota project.

Out of all of these systems, the GMD research database project Amos is the furthest along. It is the only one that provides limited real-time delivery of multimedia data.

Summary

Multimedia DBMSs can take many forms, from pseudo-repositories to advanced intelligent multimedia data management systems. We will undoubtedly see products throughout this range satisfy the requirements of many different applications. It's unlikely that every DBMS product will evolve to the high end of the spectrum in the near future. However, as multimedia data integrates into our everyday lives, most DBMS products will also evolve to become sophisticated multimedia systems with all of the features (and more) that we've discussed. MM

Acknowledgments

We would especially like to thank the companies and project teams that participated in the survey for their time, for providing the information requested, and for letting us publish it. We would also like to thank Ram Sriram (NIST) and Mark Foresti (Rome Labs) for helping fund this effort. Finally, we would like to thank Jim Richardson (Honeywell) and the past reviewers for their invaluable comments.

References

1. T.C. Rakow, E.J. Neuhold, and M. Lohr, "Multimedia Database Systems—the Notions and the Issues," *Database Systems in Office, Tech. and Knowledge*, Springer, Berlin, March 1995, pp. 1-29.
2. W. Klas and K. Aberer, *Multimedia Applications and Their Implications on Database Architectures*, GMD Tech. Report 95-20, GMD, Berlin, 1995.
3. R. Jain, ed., "NSF Workshop on Visual Information Mgt Systems," *ACM Sigmod Record*, Vol. 22, No. 33, 1993, pp. 57-75.
4. P. Pazandak, J. Srivastava, and J. Carlis, "The Temporal Component of Damsel," *Second Workshop on Protocols for Multimedia Systems (PROMS 95)*, Univ. of Salzburg, Salzburg, Austria, 1995.
5. P. Pazandak, *Multimedia Language Constructs and Execution Environments for Next-Generation Interactive Applications*, PhD thesis, University of Minnesota, Computer Science Dept. , Minneapolis, Minn., 1996.
6. R. Steinmetz and C. Engler, *Human Perception of Media Synchronization*, Tech. Report 43.9310, IBM European Networking Center, Heidelberg, Germany, 1993.
7. A. Gupta, "Visual Information Retrieval: A Virage Perspective," Virage, http://www.virage/com.
8. L. Weitzman and K. Wittenburg, "Automatic Presentation of Multimedia Documents Using Relational Grammars," *Multimedia 94: 2nd Annual Conf. on Multimedia*, ACM Press, New York, 1994.
9. T.D.C. Little and A. Ghafoor, "Interval-Based Conceptual Models for Time Dependent Multimedia Data," *IEEE Trans. on Knowledge and Data Eng.*, Vol. 5, No. 4, 1993, pp. 551-563.

10. G.D. Drapeau and H. Greenfield, "Maestro—A Distributed Multimedia Authoring Environment," *Proc. Summer 1991 Usenix Conf.*, Nashville, Tenn., 1991, pp. 315-328.
11. H. Thimm and T.C. Rakow, *Upgrading Multimedia Data Handling Services of a Database Management System by an Interaction Manager*, GMD Tech. Report 762, GMD, Berlin, 1993.
12. T.G. Kwon and S. Lee, "Data Placement for Continuous Media in Multimedia DBMS," *Int'l Workshop on Multimedia Database Mgmt. Systems (IW-MMDBMS 95)*, IEEE Computer Society Press, Los Alamitos, Calif., 1995, pp. 110-117.
13. S. Ghandeharizadeh and L. Ramos, "Continuous Retrieval of Multimedia Data Using Parallelism," *IEEE Trans. on Knowledge and Data Eng.*, Vol. 5, No. 4, Aug. 1993, pp. 658-669.

Paul Pazandak is a staff scientist at Object Services and Consulting, where he is primarily focusing on scalable software architectures for the Internet. His primary interests include object database management systems, widely distributed scalable database architectures, and distributed, multiuser Internet-based architectures. He received a BA in Physiology, and MS and PhD degrees in computer science from the University of Minnesota. He is a member of the IEEE Computer Society.

Jaideep Srivastava is currently an associate professor in the Computer Science Department at the University of Minnesota, Minneapolis. His current research is in the areas of databases, distributed systems, and multimedia computing. He received a B.Tech. degree in computer science from the Indian Institute of Technology, Kanpur, India in 1983, and MS and PhD degrees in computer science from the University of California, Berkeley, in 1985 and 1988, respectively. He is a member of the IEEE Computer Society and the ACM.

Contact Pazandak at Object Services and Consulting, 809 102nd Street East, Minneapolis, MN 55420-5409, e-mail pazandak@objs.com.