Multimedia Database Management—Requirements and Issues

Donald A. Adjeroh
The Chinese University of Hong Kong

Kingsley C. Nwosu
Lucent Technologies

The spatial, temporal, storage, retrieval, integration, and presentation requirements of multimedia data differ significantly from those for traditional data. A multimedia database management system provides for the efficient storage and manipulation of multimedia data in all its varied forms. We look into the basic nature of multimedia data, highlight the need for multimedia DBMSs, and discuss the requirements and issues necessary for developing such systems.

At the heart of multimedia information systems lies the multimedia database management system. Traditionally, a database consists of a controlled collection of data related to a given entity, while a database management system, or DBMS, is a collection of interrelated data with the set of programs used to define, create, store, access, manage, and query the database. Similarly, we can view a multimedia database as a controlled collection of multimedia data items, such as text, images, graphic objects, sketches, video, and audio. A multimedia DBMS provides support for multimedia data types, plus facilities for the creation, storage, access, query, and control of the multimedia database.

The different data types involved in multimedia databases might require special methods for optimal storage, access, indexing, and retrieval. The multimedia DBMS should accommodate these special requirements by providing high-level abstractions to manage the different data types, along with a suitable interface for their presentation.

Before detailing the capabilities expected of a multimedia DBMS and the requirements such systems should meet, we must first consider the characteristic nature of multimedia information. Then we can discuss the issues facing multimedia DBMSs.

Nature of multimedia data

We analyze the composition and characteristics of multimedia data from several perspectives. These include information overload, inadequacy of textual descriptions, multiplicity of data types, spatial and temporal characteristics, and huge volumes of data.

The integration of multimedia data types from multiple sources uniquely characterizes multimedia information systems. The data types found in a typical multimedia database include

- text;
- images: color, black and white, photographs, maps, and paintings;
- graphic objects: ordinary drawings, sketches, and illustrations, or 3D objects;
- animation sequences: images or graphic objects, (usually) independently generated;
- video: also a sequence of images (called frames), but typically recording a real-life event and usually produced by a video recorder;
- audio: generated from an aural recording device; and
- composite multimedia: formed from a combination of two or more of the above data types, such as an intermix of audio and video with a textual annotation.

Some multimedia data types such as video, audio, and animation sequences also have temporal requirements, which have implications on their storage, manipulation, and presentation. The problems become more acute when various data types from possibly disparate sources must be presented within or at a given time. Similarly, images, graphics, and video data have spatial constraints in terms of their content. Usually, individual objects in an image or a video frame have some spatial relationship between them. Such relationships usually produce some constraints when searching for objects in a database.

1070-986X/97/$10.00 © 1997 IEEE

Huge volumes of data also characterize multimedia information. For instance, to store an uncompressed image of 1024 ¥ 728 pixels at 24 bits per pixel requires a storage capacity of about 2 Mbytes. With a 20:1 compression ratio, the storage requirement could be reduced to about 0.1 Mbyte. If we consider a video example, a 10-minute sequence of the same image at 30 frames per second requires about 38,000 Mbytes of storage, reducible to about 380 Mbytes with a compression ratio of 100:1. The potential for huge volumes of data involved in multimedia information systems become apparent when you consider that a movie could run as long as two hours, and a typical video repository would house thousands of movies.

An old adage says that a picture is worth more than a thousand words. However, representing multimedia information as pictures or image sequences poses some problems for information retrieval due to the limitations of textual descriptions of a multimedia experience and the massive information available from it. The potential information overload means that users may find it difficult to make precise requests during information retrieval. The limitations of textual descriptions also imply the need for content-based access to multimedia information. Users need multiple cues (such as shape, color, and texture) that are relevant to the multimedia content.

Another characteristic of multimedia information is that interaction with such information types usually involves long-duration operations (such as with video data), and sometimes, with more than a single user (as is typical in collaborative support environments). However, in collaborative environments, it is expected that most multimedia data are likely to be accessed in a read-only mode. This assumption can be used to facilitate the provision of concurrency control algorithms.

Purpose of a multimedia DBMS

A multimedia database management system provides a suitable environment for using and managing multimedia database information. Therefore, it must support the various multimedia data types, in addition to providing facilities for traditional DBMS functions like database definition and creation, data retrieval, data access and organization, data independence, privacy, integration, integrity control, version control, and concurrency support.

The functions of a multimedia DBMS basically resemble those of a traditional DMBS. However, the nature of multimedia information makes new demands—including determining what is needed and how to provide that functionality.

Using the general functions provided by a traditional DBMS as a guide, we can describe the purposes of a multimedia DBMS as follows:

- *Integration*. Ensures that data items need not be duplicated during different program invocations requiring the data.
- *Data independence*. Separation of the database and the management functions from the application programs.
- *Concurrency control*. Ensures multimedia database consistency through rules, which usually impose some form of execution order on concurrent transactions.
- *Persistence*. The ability of data objects to persist (survive) through different transactions and program invocations.
- *Privacy*. Restricts unauthorized access and modification of stored data.
- *Integrity control*. Ensures consistency of the database state from one transaction to another through constraints imposed on transactions.
- *Recovery*. Methods needed to ensure that results of transactions that fail do not affect the persistent data storage.
- *Query support*. Ensures that the query mechanisms are suited for multimedia data.
- *Version control*. Organization and management of different versions of persistent objects, which might be required by applications.

In concurrency control, a transaction is a sequence of instructions executed either completely or not at all. In the latter case, the database is restored to its previous state. Defining the appropriate granularity for concurrency is a problem in multimedia databases. Traditional databases use a single record or table as the unit of concurrency; multimedia databases typically use a single object (or composite object) as the logical unit of access. Thus the single multimedia object could form the unit of concurrency.

In achieving persistence, a simple method is to

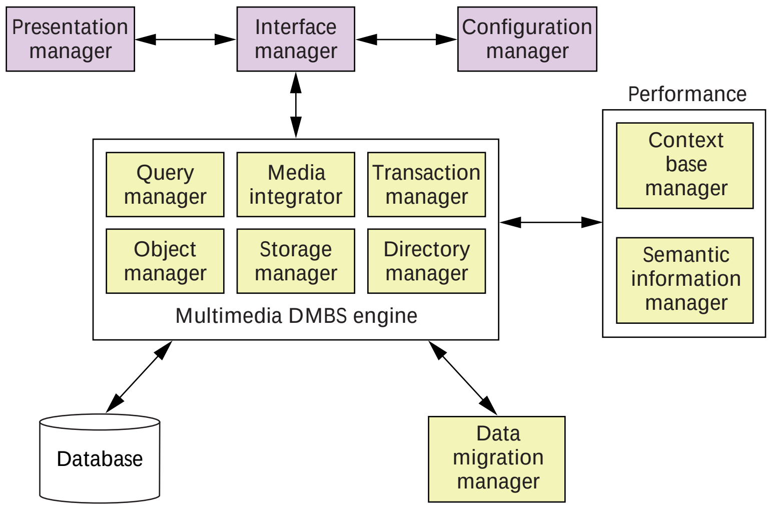


Figure 1. A high-level architecture for a multimedia DBMS that meets the requirements for multimedia data.

store the multimedia files in some operating system files. However, the huge data volumes make this approach costly to implement. Moreover, the system also needs to store the multimedia metadata and possibly composite multimedia objects. Thus, most multimedia DBMSs classify the data as either persistent or transient and store only persistent data after transaction updates. Transient data are used only during program or transaction execution and are removed afterwards.

Traditionally, a query selects a subset of the data objects based on the user's description (usually some form of query language) of what data to access. A query usually involves various attributes, possibly keyword-based or content-oriented, and is usually interactive. Thus, functions for relevance feedback and query formulation, similarity (rather than exact) matches, and mechanisms for displaying ranked results are important in a multimedia DBMS.

Version control becomes important when a persistent multimedia object is updated or modified, as some applications might need to access previous states of the object. A DBMS provides such access through versions of the persistent objects. For a multimedia DBMS, the huge volumes of data reinforces the importance of efficiently organizing such versions. Moreover, the available storage might limit the provision of versions. In addition, version management may involve not only versions of single objects, but also versions of the complex objects that make up the multimedia database.

The special nature of multimedia data also makes it important to support new special functions. These include object composition and decomposition, management of huge volumes of multimedia data, effective storage management, and information retrieval and handling of spatial and temporal data objects. Khoshafian and Baker[1] and Kim[2] provide more detailed discussions of some of these issues.

Requirements for the multimedia DBMS

For the multimedia DBMS to serve its expected purpose, it must meet certain special requirements. Khoshafian and Baker[1] described a multimedia DBMS architecture and the interaction of the different components needed to provide the services expected. The requirements are divided into the following broad categories:

- Traditional DBMS capabilities
- Huge capacity storage management
- Information retrieval capabilities
- Media integration, composition, and presentation
- Multimedia query support
- Multimedia interface and interactivity
- Performance

In addressing these requirements when building a multimedia database system, one must also address several other questions to achieve full functionality, including

- How to build a multimedia database system that encompasses several application domains (that is not restrictive in terms of its domain applicability)?
- What are the levels of granularity for information decomposition, storage, and management? And how the underlying techniques and structures can be mapped and used on the units of data?
- Knowing the data compositions of a multimedia database, how can one reliably and efficiently develop a query language that supports the myriad access methods associated with and necessary for the diverse object types? How will the query language support the multimedia data's different characteristics and morphologies?

- What kind of presentation infrastructure will the multimedia system have to accommodate the diverse presentation requirements and modes for the different multimedia data? How can one synchronize presentations to support the temporal and spatial requirements of the different multimedia data?

- Given that different media types have differing modification and update requirements, how will the system update different components of the multimedia session? What levels of granularity will those updates have?

Figure 1 shows a sample high-level architecture for a multimedia DBMS that addresses some of the requirements that have been discussed.[3] This configuration includes most of the management modules associated with a traditional DBMS. In addition, it contains some of the modules that are required specifically for multimedia data management, such as the media integrator and object manager. However, most of the additions to the traditional DBMS are external to the core of the multimedia DBMS. These include the presentation, interface, and configuration managers. The configuration also includes a context-base and semantic information manager, which are part of the performance module.

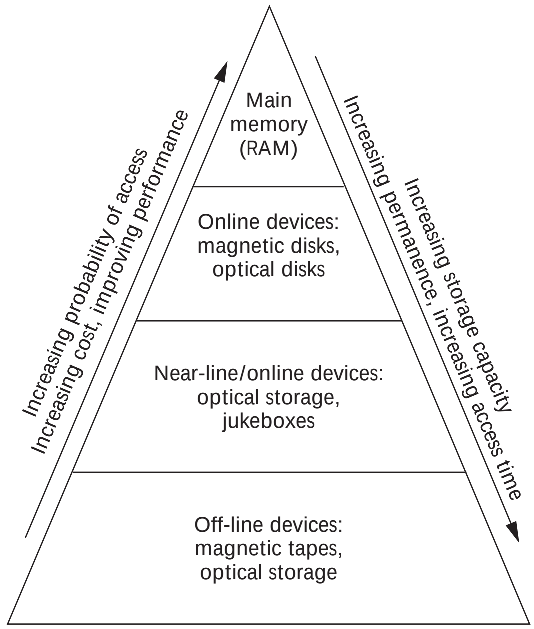


Figure 2. Hierarchically organized storage for multimedia databases.

Huge capacity storage management

The storage requirements in multimedia systems can be characterized by their huge capacities and the storage system's hierarchical (pyramidal) organization (see Figure 2). Hierarchical storage places the multimedia data objects in a hierarchy of devices, either online, near-line, or offline. In general, the highest level provides the highest performance, highest cost, smallest storage capacity, and least permanence. Note, however, that permanence improves—at significant additional cost—with the use of nonvolatile random access memory.

Another unique use of this hierarchical storage organization is that the higher levels of the hierarchy can be used to store smaller abstractions (or representations) of the actual multimedia data, which can be used to facilitate faster browsing and previewing of the database content.

Cost and performance (in terms of access time) decrease as we go down the hierarchy (pyramid), while storage capacity and permanence increase. Typically, in most multimedia storage systems the highest level of storage is (volatile) random access memory, followed by magnetic disk drives. These provide online services. Optical storage devices provide the next level of storage. Online in some cases, they are near-line (like jukeboxes) in most cases. The lowest level in the storage hierarchy represents offline storage devices, including magnetic tapes, optical disks, and so forth. These may or may not be directly connected to the computer. They offer the highest storage capacity and permanence but provide the least performance in terms of access time.

A multimedia DBMS must therefore manage and organize multimedia data stored at any level in the hierarchy. It must have mechanisms for automatically migrating multimedia data objects from one level of the storage hierarchy to another. A detailed treatment of these migration policies exceeds our scope here, but they must be based on some clearly defined criteria such as frequency of use. In general, even when the data is stored in offline storage devices, the multimedia DBMS should have information on how to easily locate the specific device containing the multimedia data being sorted.

Data migration in multilayered storage systems is not peculiar to multimedia DBMSs. All databases handling huge amounts of data must address this issue. The interconnection between the memory systems is obviously a problem, especially when a multimedia database involves distributed sources of data. As a result, the problem of data migration may require the consideration of other network related issues, such as data availability rates, bandwidth limitations, and network delays.

Query support and information retrieval

Querying in multimedia databases can involve different multimedia data types, keywords, attributes, content, or even contextual information. Because of the different ways in which users think about multimedia data, multimedia query can simultaneously involve multiple cues, necessitating multiple or multidimensional indices. Queries are usually imprecise. Because of this and the difficulty of ensuring exact matches between multimedia data items, retrieval usually involves comparing data items for similarity or partial (rather than exact) matching. Thus, since queries might not yield exact matches, we need facilities for ranking the retrieved results according to how closely they match the given query. Similarly, we should have methods to prune results that do not seem to satisfy the query. Doing so reduces the potentially enormous computation needed for further matching.

With the ranking, the multimedia DBMS should also support browsing the various retrieved items. We might also want to retrieve similar items based on one or more of the already retrieved items.

With users unsure of the information sought, a true multimedia DBMS also needs a facility to support incomplete information. More importantly, since the information extracted to index the multimedia data or from the user query might contain errors, query interpretation should provide for uncertainties in the information. This might require an iterative search mechanism and a relevance feedback mechanism along with techniques for query reformulation.

Media integration, composition, and presentation

Given the multiplicity of data types supported, the multimedia DBMS should also provide facilities for integrating data items (from possibly disparate media types) to form new composite multimedia types and for presenting such data at a given site within the required time frame. Multimedia integration, composition, and presentation are exacerbated by the often continuous (temporal) nature of multimedia data—especially video, animation, and audio. Moreover, certain applications, such as geographic information systems, may require a multimedia DBMS to address spatial information. All these factors put together make multimedia composition and presentation a complex process that the multimedia DBMS must support to meet the diverse user community's needs.

The integration problem can be ameliorated in some cases, especially where the multimedia database system is tailored to the requirements of some target user community. In such special cases the multimedia DBMS could support specific features not needed for other applications.

Multimedia interface and interactivity

The diverse nature of multimedia data calls for an equally diverse interface for interacting with the database. Typically, each media data type has its own method for access and presentation. For instance, video and audio data will need different user interfaces for presentation and query. For some multimedia applications, especially those involving continuous media, the user often expects the interactive facilities of a VCR or tape recorder, such as fast forward and reverse. When a multimedia system provides such services, it has implications for the database, especially retrieval of the needed multimedia objects, their integration, and their synchronization. Thus the multimedia DBMS might need to support such forms of interactivity.

Performance

Efficiency is an important consideration in a multimedia DBMS. Multimedia database systems make new performance demands on media access, storage, indexing, retrieval, and query optimization. The different data types involved in multimedia databases might also require special methods for optimal storage, access, indexing, and retrieval. Rodriguez[4] discusses some of the performance requirements that must be considered for multimedia DBMSs. These include efficiency, reliability, real-time execution, guaranteed and synchronized delivery of multimedia presentations, and quality-of-service (QoS) acceptable to the users. We will further discuss some of the issues affecting performance later.

Issues in multimedia DBMSs

To meet the requirements discussed, the multimedia DBMS must address a number of issues, including

- Multimedia data modeling
- Multimedia object storage
- Multimedia integration, presentation, and QoS
- Multimedia indexing, retrieval, and browsing
- Multimedia query support

- Distributed multimedia database management
- System support

Multimedia data modeling

Data models are central to multimedia database systems. A data model must isolate users from the details of storage device management and storage structures. It requires the development of appropriate data models to organize the various data types typically found in a multimedia database system.

Multimedia data models (just like traditional data models) capture the static and dynamic properties of the database contents, and thus provide a formal basis for developing the appropriate tools needed in using the multimedia data. The static properties could include the objects that make up the multimedia data, the relationships between the objects, the object attributes, and so on. Examples of the dynamic properties include interaction between objects, operations on objects, user interaction, and so forth.

However, the unique nature of multimedia data requires certain new considerations when choosing the data model. For instance, some multimedia data types (such as video) or group of types (example, video and images) might require special data models for improved modeling efficiency and flexibility. Moreover, the importance of interactivity in multimedia systems makes their support by the data model an important issue. Furthermore, it may be necessary to consider new integrity constraints in the context of multimedia databases.

Various data models, such as network, relational, semantic, and object-oriented models already exist for traditional databases,[5] and a few have been proposed for multimedia databases.[6,7] Two basic approaches have been used in modeling multimedia data. The first involves building a multimedia data model on top of an underlying traditional database data model (usually relational or object-oriented databases) by using appropriate interfaces for the multimedia data. The problem with this approach is that the underlying structures are not designed for multimedia data. Often, the significant differences between the requirements of the traditional and multimedia data make the interface a bottleneck in the overall system.

These problems led to the second method, which opts to develop true multimedia-specific data models from scratch, rather than on top of an existing traditional database system. Nonetheless, a consensus almost exists that such efforts should be based on object-oriented techniques. Current issues include developing appropriate data models for individual multimedia data types (such as video, images, or visual data), uniform modeling of arbitrary data types, and supporting huge volumes of multimedia data, multimedia interactivity, and content-based information using these models. Some authors have gone so far as to claim that the data model for a multimedia DBMS can only be fully achieved by object-oriented technology.[8-10]

Data compression schemes, in combination with the data transformation, help to reduce the huge capacity requirements.

Multimedia object storage

Physically storing multimedia data requires methods for transforming, managing, transferring, and distributing huge volumes of data. Typical multimedia systems use a hierarchy of storage devices. Online high-speed devices (such as random access memories) and magnetic disks store multimedia data currently being used, while offline, low-speed devices (like optical storage and tapes) store long-term archival data. Performance then depends on the efficiency of the migration mechanisms used to assign the multimedia data items to the optimal level in the storage hierarchy.

Data compression schemes, in combination with the data transformation, help to reduce the huge capacity requirements. The basic method here is to transform the multimedia data to some transform space to remove the redundancies in the original data. Coding schemes code the transformed data for storage or transmission. Decompression is accomplished from the reverse process of decoding and re-transforming the data into its original form. This process often involves some loss of data, which a majority of multimedia applications can tolerate.

The huge volumes of data often involved and the constraints certain multimedia data types impose on the presentation make multimedia object storage a major consideration in database issues. Depending on the level of granularity, a multimedia object can represent the entire video sequence for a movie, a subsequence from the video, a single frame or image, or even individual objects in the image or video frame.

The major issues here are the limited available storage, the bandwidth limits of the storage system and communication channel, and the multimedia data type's availability rates. The data availability rate[11] indicates the minimum amount of data required per unit time to meet acceptable levels of quality during presentation of the multimedia object. Viewed from this standpoint, multimedia data's storage requirements are most susceptible to decomposing the data into smaller multimedia objects. Each smaller object can be stored in the smaller available storage units.

As a necessary condition for storage allocation, at presentation time the data from the different storage units, when combined together, meet the data availability rates of the given multimedia data type. With the hierarchical storage arrangement, multimedia objects can be stored at different levels. As the utility rate of multimedia data objects changes, such objects will need to be reallocated, possibly to different storage devices on different levels of the storage hierarchy. The problems then involve finding optimal methods for multimedia object decomposition, allocation, and reallocation.

Multimedia integration, presentation, and QoS

Unlike traditional data, multimedia data have presentation constraints. These mainly result from the continuous nature of some multimedia data types, which requires presenting certain amounts of data within a given time for the presentation to seem natural to the user. When multimedia data are distributed and transported over networks, the problems of presentation become even more acute. Here, one can easily experience network problems, such as limited bandwidth and statistical network delays.[12]

Continuous media by definition are time-dependent, so timing becomes an important factor in their delivery and presentation. Therefore, in multimedia DBMSs the response to a query is often judged by both the correctness and the quality of the retrieved results.

From the user's point of view, the QoS parameter specifies, qualitatively, the acceptable levels of performance for the various services provided by the multimedia system and may affect the results of the multimedia presentation. Thus, to support multimedia presentations where a user can specify various QoS levels for different services, the multimedia DBMS must support the specified QoS levels and a QoS management service. This typically involves providing an appropriate mapping from the user's QoS to the system's QoS and vice versa.

When presenting different types of multimedia data—such as video and audio—together, problems of media integration and synchronization also become important. The multimedia DBMS must provide a mechanism to ensure good synchronization of the presented data while still meeting other requirements such as the data availability rates and the QoS. In some situations, the multimedia DBMS may have to rely on an explicit synchronization manager to ensure synchronization within a given data type and between different data types.

Multimedia indexing

As in traditional databases, multimedia information can be retrieved using identifiers, attributes, keywords, and their conjunctions using conditional statements. Keywords are by far the predominant method used to index multimedia data. A human typically selects keywords from a set of specialized vocabulary. While simple and intuitive, this method usually creates problems when applied to multimedia data: it is basically manual and time consuming, and the resulting indices are highly subjective and limited depending on the vocabulary.

Another method, content-based access, refers either to the actual contents of the multimedia database or to derived contextual information. Intensive research has focused on content-based indexing in recent years, with the goal of indexing the multimedia data using certain features derived directly from the data. Various features, such as color, shape, texture, spatial information, symbolic strings, and so on, have been used to index images.

Deriving such features requires automatic analysis of the multimedia data. The primary methods used for image and video data are image processing, image understanding, and video sequence analysis. With video data, the video sequence is first separated into its constituent scenes, then representative abstractions (usually key frames) are selected to represent each scene. Further indexing on the video is based on the key frame, as in the case for images.

For audio data, content-based indexing could involve analysis of the audio signal or automatic speech recognition followed by keyword-based indexing. On the other hand, indexing can be based on other information depending on the type of audio data. For example, some developers have used rhythm signature, chord, and melody for content-based indexing of music data.[13]

Similarly, methods for content-based search and retrieval of audio data have been proposed based on the characteristics of audio data, as indicated by its perceptual and acoustic features.[14]

Using content-based indexing implies the consideration of certain issues. First, the same multimedia data could mean different things to different people. Second, users typically have diverse information needs. Thus, it is evident that a single feature may not be sufficient to completely index a given multimedia data type. Therefore, it becomes difficult to identify the features that are most appropriate in any given environment.

Another problem has to do with efficiency: making the indexing fast and storing the indices efficiently for easy access, since multimedia data typically come in huge volumes. Because of the diverse content inherent in multimedia data, indexing has not been completely automated. For example, while the computer can easily analyze a picture containing works of art, it is almost impossible for the computer to automatically determine the meaning of the art object. Only a human can provide such information.

Multimedia query support, retrieval, and browsing

User queries are often processed using only available indices. However, unlike in traditional databases, matches in multimedia queries are not exact matches. Often when comparing two multimedia data items, approximate or similarity matches result. Given that various items can resemble the same input data, a single query might yield many items in response.

Various research efforts have chosen to investigate issues on similarity matching involving multiple indices and ranking. Also being developed are appropriate ways of presenting the retrieved information, such as through a browsing interface. A user-directed browsing lets the user retrieve any information potentially related to the current results by selecting the data items for further consideration.

Among the issues involved in multimedia query support is the availability of a multimedia query language capable of supporting both the various media types encountered in a typical multimedia database and new requirements such as fuzzy query predicates. Such query models should also provide mechanisms for users to reformulate their queries, perhaps based on the already retrieved results.

Query-by-example is the primary method used to enter queries in multimedia databases, especially in those involving images. Here, the user makes a request using an existing example (for example, similar images). Thus, the interface used to enter the query into the system becomes an issue. Since different multimedia data types may require different query interfaces, the problems to consider include how to integrate the various interfaces in an integrated multimedia database system. Other problems to be resolved include querying spatial data and content-based video query, which could involve temporal and spatial information.

Query-by-example is the primary method used to enter queries in multimedia databases, especially in those involving images.

Distributed multimedia database management

Distributed multimedia DBMS loosely refers to a collection of various (possibly) independent multimedia database management systems, located in disparate locations, that can communicate and exchange multimedia data over a network. Multimedia systems are usually distributed in the sense that a single multimedia interaction often involves data obtained from distributed information repositories. This is typically the case in collaborative multimedia environments, where multiple users in possibly disparate physical locations manipulate and author the same multimedia document. Moreover, issues like storage problems and data generation may also force multimedia system designers to place multimedia data in different physical locations.

To support the information required in such distributed and collaborative environments, a distributed multimedia DBMS must address the general problems in distributed databases, such as distributed and parallel query processing, distributed transaction management, data location transparency, data security, and so forth. In addition, network issues such as limited bandwidth and network delays become important considerations, since they could have adverse effects on the QoS supported.

Unlike in the traditional DBMS, data replication is often not encouraged in a distributed multimedia DBMS due to the huge data volumes. The client-server computing model, in which a server application services multiple client applications—with the clients and server residing in possibly different machines—has proven suitable for multimedia systems in general and distributed multimedia DBMSs in particular.

System support

Multimedia applications in general, and distributed multimedia database systems especially, raise new issues in all aspects of the computer system, from operating systems to networks to general hardware. Most generally available operating systems do not support real-time operations adequately. Rather, they provide hardware front-ends for transmitting and presenting multimedia data.

Some multimedia data, such as continuous media, may require real-time delivery and presentation, although the real-time requirements might not be as stringent as those encountered in hard real-time systems. Thus, the multimedia database system cannot fully provide its functionalities until support for real-time continuous media data becomes an integral part of the operating system. Efforts on various fronts have concentrated on this problem, including research on resource scheduling, operating system support for QoS, use of multilevel and user-level threads, and so on.

Other characteristics of multimedia, such as the huge data volumes, may mandate special constraints on the system in terms of memory management, CPU performance, throughput, and so forth. Related issues include general considerations on I/O hardware to support the various media types involved in multimedia databases. Communication networks—needed to transport the data for distributed multimedia environments—must support bandwidth and delay guarantees as needed to meet the stringent QoS requirements for certain multimedia applications.

Applications

In general, multimedia database management systems applications can be found wherever there is a need to manage multimedia data cost-effectively. Thus multimedia DBMSs have found applications in such diverse areas as education (digital libraries, training, presentation, distance learning), healthcare (telemedicine, health information management, medical image systems), entertainment (video-on-demand, music databases, interactive TV), information dissemination (news-on-demand, advertising, TV broadcasting), and manufacturing (distributed manufacturing, distributed collaborative authoring). Other areas may include finance, video conferencing, electronic publishing, electronic commerce, and geographic information systems.

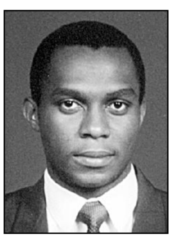
A number of multimedia DBMSs already exist. Most are extensions from existing object-oriented or relational DBMSs. The capabilities of existing multimedia DBMSs can be evaluated by the extent to which they support different media types (especially image and video). They can also be evaluated by their ability to support special functionalities required of a database system to manage multimedia data, such as real-time delivery and content-based query and retrieval.[15]

Unfortunately, most existing multimedia DBMSs fall short. However, rigorous research is being geared toward the various parts of the problem, and we expect new systems with better capabilities in the near future. Future trends include performing indexing, retrieval, and browsing directly on the compressed data, especially for video data; video data management; multimedia query language; uniform indexing frameworks for the different data types; content-based image and video retrieval; and multimedia transport and delivery over the Internet. MM

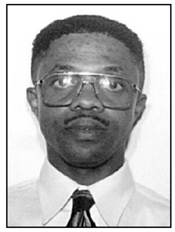
References

1. S. Khoshafian and B.A. Baker, *Multimedia and Imaging Databases*, Morgan Kaufmann Publishers, San Francisco, Calif., 1996.
2. W. Kim, ed., *Modern Database Systems: The Object Model, Interoperability, and Beyond*, Addison-Wesley, Reading, Mass., 1995.
3. A. Narasimhalu, "Multimedia Databases," *Multimedia Systems*, Vol. 5, No. 4, Oct. 1996, pp. 226-249.
4. A.A. Rodriguez and L.A. Rowe, "Multimedia Systems and Applications," *Computer*, Vol. 28, No. 5, 1995, pp. 20-22.
5. R. Hull and R. King, "Semantic Database Modeling: Survey, Applications, and Research Issues," *ACM Computing Surveys*, Vol. 19, No. 3, 1987, pp.201-260.
6. C.Y.R. Chen et al., "Design of a Multimedia Object-Oriented DBMS," *Multimedia Systems*, Vol. 3, No. 5-6, 1995, pp. 217-227.
7. S. Marcus and V.S. Subrahmanian, "Foundations of Multimedia Database Systems," *J. ACM*, Vol. 43, No. 3, May 1995, pp. 474-523.
8. D. Woelk, W. Kim, and W. Luther, "An Object-Oriented Approach to Multimedia Databases," *Proc. ACM SIGMOD 1986 Int'l Conf. on Management of Data*, Vol. 15, No. 2, 1986, ACM Press, New York, pp. 311-325.
9. D. Woelk and W. Kim, "Multimedia Information Management in Object-Oriented Database Systems," *Proc. Int. Conf. on VLDB*, Brighton, England, Sept. 1987, pp. 319-329.

10. Y. Masunaga, "Multimedia Databases: A Formal Framework," *Proc. IEEE CS Office Automation Symp*., IEEE Computer Society Press, Los Alamitos, Calif., Apr. 1987, pp. 36-45.
11. C. Orji, K. Nwosu, and N. Rishe, "Multimedia Object Storage and Retrieval," *Proc. Int'l Symp. on Multimedia Systems*, Information Processing Society of Japan, Yokohama, Japan, 1996, pp. 368-375.
12. D.A. Adjeroh and M.C. Lee, "Synchronization and User Interaction in Distributed Multimedia Presentation Systems," *Multimedia Database Systems: Design and Implementation Strategies* , K.C. Nwosu, B. Thuraisingham, and P.B. Bera, eds., Kluwer Academic Publishers, Boston, Mass.,1996, pp. 251-297.
13. T-C. Chou, A.L.P. Chen, and C.C. Liu, "Music Databases: Indexing Techniques and Implementation," *Proc. 1996 Int'l Workshop on Multi-Media Database Management Systems*, IEEE Computer Society Press, Los Alamitos, Calif., 1996, pp. 46-53.
14. W. Wold et al., "Content-Based Classification, Search, and Retrieval of Audio," *IEEE MultiMedia*, Vol. 3, No. 3, 1996, pp. 27-36.
15. P. Pazandak and J. Srivastava, "Metrics for Evaluating ODBMS Functionality to Support MMDBMS," *Proc. Int'l Workshop on Multi-Media Database Management Systems*, IEEE Computer Society Press, Los Alamitos, Calif., 1996, pp. 72-79.



Donald A. Adjeroh is a post-doctoral fellow in the Department of Computer Science and Engineering of the Chinese University of Hong Kong. He holds a BEng degree (First Class Honors) in electronics and computer engineering, obtained in 1990, and an MS in computer engineering, obtained in 1994, and a PhD in computer science from the Chinese University of Hong Kong, obtained in 1997. Adjeroh's current research focuses on indexing and retrieval of visually stimulated information. His general research interests are in video processing, computational aspects of vision, distributed multimedia systems, digital libraries, and knowledge based systems. He is a member of IEEE Computer Society.



Kingsley C. Nwosu is a member of technical staff with Lucent Technologies, Whippany, New Jersey. He is also an adjunct professor in the Department of Mathematics and Computer Science at Rutgers University, and a visiting professor in the Department of Mathematics and Computer Science at Montclair State University, New Jersey. His current areas of research include multimedia information and computing systems, object-oriented programming and design, distributed processing and computing, information storage and retrieval, and parallel algorithm design and analysis.

Nwosu holds a BS degree in computer science from the University of Southwestern Louisiana, Lafayette, an MS in computer science from the University of Oklahoma, Norman, and a PhD in computer science from Syracuse University, Syracuse, New York. He served as program chair for the 1994, 1995, and 1996 IEEE International Workshop on Multimedia Database Management Systems. He was also a guest editor for a special issue of the *Journal of Multimedia Tools and Applications* on multimedia database systems. He recently co-edited a book on multimedia database systems. He is a member of ACM, IEEE Computer Society, IEEE Communication Society, New York Academy of Science, and National Council on Systems Engineering.

Readers may contact Nwosu at Lucent Technologies, Network Systems, Room 2E-223A, 67 Whippany Rd., Whippany, NJ 07981-0903, e-mail knwosu@lucent.com.