A Digital On-Demand Video Service Supporting Content-Based Queries[1]

T.D.C. Little, G. Ahanger, R.J. Folz, J.F. Gibbon, F.W. Reeve, D.H. Schelleng, and D. Venkatesh

Multimedia Communications Laboratory
Department of Electrical, Computer and Systems Engineering
Boston University, Boston, Massachusetts 02215, USA
(617) 353-9877, (617) 353-6440 fax
tdcl@bu.edu

MCL Technical Report 08-01-1993

Abstract–Video-on-demand represents a key demonstrative application for enabling multimedia technology in communication, database, and interface research. This application requires solving a number of diverse technical problems including the data synchronization problem for time-dependent data delivery.

In this paper we describe the general requirements of video-on-demand and introduce a system supporting content-based retrieval and playback for the structure and content of digital motion pictures. In our model we capture domain-specific information for motion pictures and provide access to individual scenes of movies through queries on a temporal database. We describe our implementation of this service using existing workstation and storage technology.

Keywords: Multimedia databases, video-on-demand, applications, temporal data management, content-based retrieval.

[1]In *Proc. 1st ACM Intl. Conf. on Multimedia*, Anaheim CA, August 1993, pp. 427-436. This work is supported in part by the National Science Foundation under Grant No. IRI-9211165.

1 Introduction

Future multimedia information systems will have a dramatic effect on the dissemination of information to individuals. These systems will provide a plethora of services including games, movies, home shopping, banking, health care, electronic newspapers/magazines, classified advertisements, etc. One manifestation of this view is the development of systems supporting video-on-demand (VOD). In a VOD system, video data, including motion pictures, are provided to a user community on an individual basis. Fig. 1 shows an example of a graphical user interface to a VOD system.

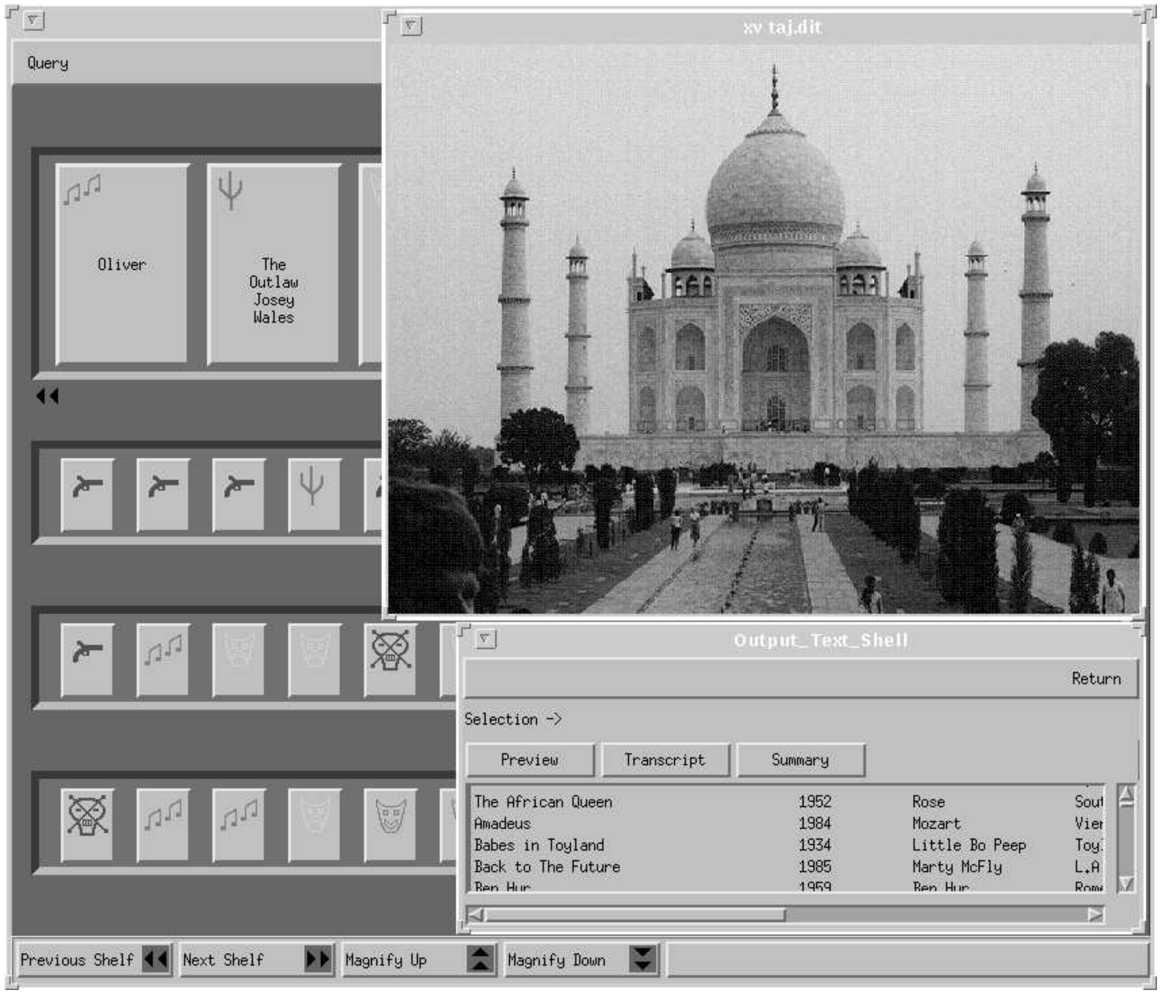
A VOD system must provide mechanisms to access enormous amounts of information, especially as derived from video data. This implies the existence of very large databases and the means of accessing them. On the other hand, individual users require the ability to filter unwanted information in the selection of appropriate programming.[2]

We envision a VOD system that allows the viewing preferences of each individual to be tailored and adapted to the available programming (e.g., one viewer's interest in classic movies, another's restriction to children's programming). Such a system would filter both stored and "live" television broadcasts and would offer these selections to the viewer instead of the plethora of choices possible. Other features would permit more fine-grained information filtering. For example, specific topics of interest could be identified within a set of newscasts or documentaries.

In order to support this vision, we require identification of the content of unstructured audio and video data. In the case of a newscast, the content is readily available from its producer. For motion pictures the content is embedded in the images themselves (in the absence of a script). We ultimately want to provide interesting, useful, and diverse means of accessing such multimedia information from a very large realm of data. This requires searching for specific items in unstructured data (e.g., "find all scenes of a movie with a certain character"). To provide this type of access, we need to model the content of the data when data can be unstructured.

In addition to the modeling of data and queries, we also need the ability to extract data to fit these models from the unstructured data. This requires identifying and classifying information from images, sound and text. Furthermore, timing information is required to support audio and video data delivery, playout, and queries based on temporal ordering.

[2]Cable TV systems with more than 500 channels are proposed [2].



```
The African Queen            1952        Rose              Sou
Amadeus                      1984        Mozart            Vie
Babes in Toyland             1934        Little Bo Peep    Toy
Back to The Future           1985        Marty McFly       L.A
Ben Hur                      1959        Ben Hur           Rom
```

Figure 1: A User Interface to VOD Services

Our objective in this work is to demonstrate the utility of our models for time-dependent multimedia data [19] in the construction of a VOD system. In this process we have sought to develop interesting access approaches to the retrieval of motion pictures. The prototype system, called the Virtual Video Browser (VVB), is designed to support *temporal access control* operations based on data structures describing the digital movies. Content-based retrieval is achieved by using a domain-specific model for motion pictures. The VVB application characterizes motion pictures and their attributes to the level of movie *scenes* by using a movie-specific data schema. This schema interfaces to our temporal model to provide temporal access control (TAC) operations such as fast-forward and reverse playout.

Research related to this effort includes modeling of unstructured data for content-based retrieval [26, 25, 39, 41, 42, 46], modeling of the temporal component of multimedia data [8, 9, 19, 16], modeling of application-specific multimedia data (movies) [11, 23, 24, 31, 35, 36], systems support for delivery of audio and video and video servers [1, 29, 43], and network-based VOD system design [14, 13, 27, 33, 34].

The remainder of this paper is organized as follows. In Section 2 we overview VOD network services. In Section 3 we describe domain-specific attributes of motion pictures and our base temporal modeling facility. Section 4 describes the design and operation of the VVB system. Section 6 concludes the paper.

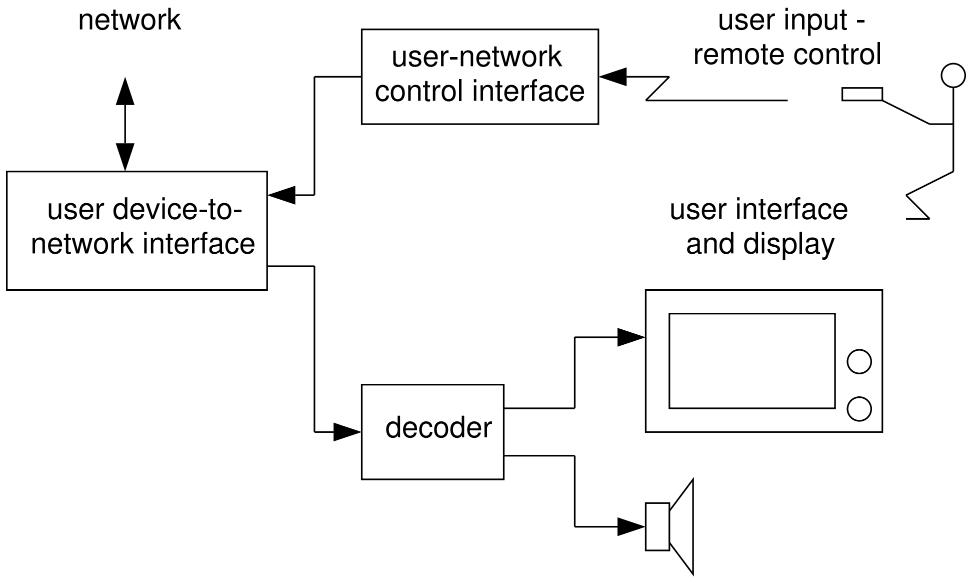

network
user-network control interface
user input - remote control
user device-to-network interface
user interface and display
decoder

Figure 2: CPE for VOD

2 VOD Network Services

Video-on-demand promises tremendous change in information distribution. Envisioned features of the simplest systems provide services to allow selection, delivery and viewing of movies through interaction with a viewing device such as a television with remote remote control. This type of device, or customer premises equipment (CPE), is illustrated in Fig. 2 (adapted from [13]).

VOD differs from existing video rental services in its potential to deliver many additional services such as interactive learning and information retrieval. In its basic form, a VOD interaction scenario consists of a movie database perusal or query, a request for a feature presentation, and movie interaction through TAC operations. Current VOD systems are described by a scale of interaction capability as follows [13]:

- Pay-Per-View (PPV)
- Quasi Video-On-Demand (Q-VOD)
- True Video-On-Demand (VOD)

PPV represents an incremental change from broadcast television supporting prescheduled programs selected by a user, and is easily supported by simple VCRs and network communication. Q-VOD provides additional interaction by grouping requests for individual programs and scheduling them at regular intervals. This provides the ability to pause a program and resume by restarting the program within a separate group at a constant time offset. This scheme requires an independent data source for each group request. True VOD does not have these limitations and assumes total interaction capability. It is also the most expensive to implement with current technology as each user can require a unique data source.

A typical architecture for VOD services is shown in Fig. 3 (adapted from [21]). Here, a set of video databases are interconnected with a set of servers which perform routing of VOD traffic to individual users. Users connect to the VOD service through access points typical of a central office (CO) exchange. It is envisioned that each server manages connection-oriented VOD traffic for multiple sessions.

The requirements of a VOD service are primarily data transport, data packaging, and billing, but also include the provision of *information warehouses (IW)* [13]. These IWs can be part of the VOD service or be complementary, provided by information vendors.

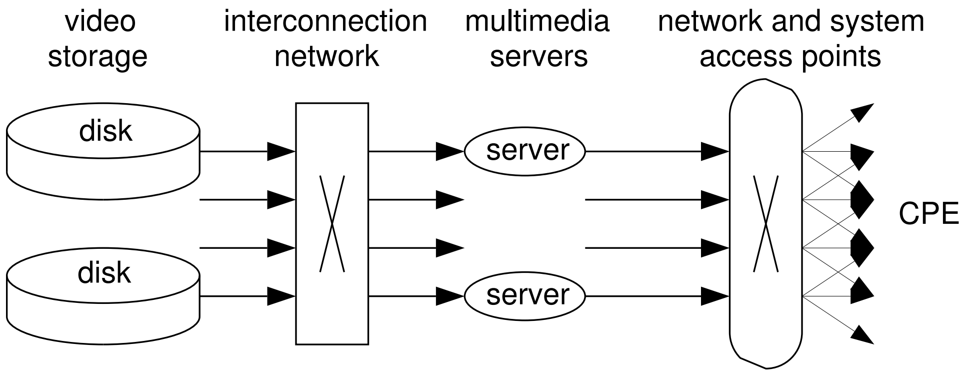


Figure 3: VOD Architecture

A typical storage requirement for a VOD system depends on the data encoding format. If data modification by the user is not anticipated for this service, an asymmetrical video coding scheme can be used such as proposed by MPEG [40]. For an asymmetrical scheme, compression is complex and costly yet yields better compression ratios than a symmetrical one (e.g., JPEG). At a compression yielding approximately 1.5 Mbit/s for 90 minute movies, a 50 GByte jukebox can hold about 50 movies. To ameliorate the problem of supporting access to every movie ever produced, access to movies can be prioritized using a storage hierarchy and assignment of a movie demand function based on movie popularity [27, 34]. Using this scheme, movie data are replicated based on anticipated user demand or are removed from fast, on-line storage yet are still accessible with additional latency.

Recent work in the development of VOD architectures includes that of Gelman et al. [14, 13], Sincoskie [34], Ramarao and Ramamoorthy [27], and Silver and Singh [33]. Commercial applications of VOD are currently being pursued by numerous organizations.

3 VOD Databases

A VOD database server is required to support time-dependent data delivery to a large number of individuals. The design of such a system must consider the impact of time-dependent data support over long-lived connections. To support these data, a multimedia information system must capture data timing requirements using an appropriate data structure suitable for data playout and data evolution due to editing. In this section we begin with the domain-specific conceptual model for motion pictures and then consider temporal and physical models for supporting time-dependent audio and video data.

3.1 Modeling of Motion Pictures

A motion picture modeled as data consists of a finite-length of synchronized audio and still images. This model represents a considerable simplification of the possible heterogeneous objects in a general multimedia data model which can be comprised of arbitrarily linked data types. To provide useful access functionality, this model is extended to capture movie *content*. *Context* can also be captured based on relationships among movie components [11]. Davenport et al. [11] describe the fundamental film component as the *shot*: a contiguously recorded audio/image sequence. To this basic component, attributes such as content, perspective, context can be assigned, and later used to formulating a specific view on a collection of shots (i.e., a movie generated by query). This approach of tailoring the output to the user is also emphasized by Lippman and Bender [18] Bender et al. [3], and by Loeb [23, 24].

In related work, Aguierre Smith and Davenport [35, 36] use a technique dubbed *stratification* for aggregating collections of shots by contextual descriptions called *strata*. These strata provide access to frames over a temporal span rather than to individual frames or shot endpoints. This technique can then be used primarily for editing and creating movies from source shots. The current implementation of this work requires manual semantic analysis to generate the contextual information. With such a technique it might be possible to present a query to generate a movie with a “happy ending” rather than another possible outcome. Such a scheme has been implemented by Sasnett [32] for interactive movies using the hypertext paradigm.

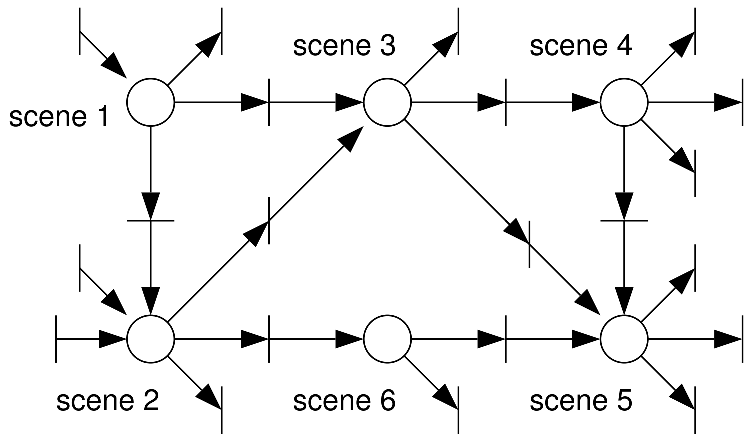


Figure 4: TPN for Representing Relationships Among Scenes of a Motion Picture

An alternate paradigm for playout of digital movies is the concurrent delivery of scenes instead of the nominal singular and sequential nature of conventional movies. For example, a murder-mystery (or sporting event) might be visualized though two concurrent displays, providing a perspective of both a detective and criminal simultaneously. Both concurrency and alternate sequence selection are representable using the timed Petri net (TPN) and the hypertext paradigm [38, 22] (e.g., Fig. 4), and can be supported in a VOD database.

3.2 Information Extraction from Unstructured Data

In a conventional DBMS (e.g., using a relational model), access to data is based on distinct attributes for well-defined data developed for a specific application. For unstructured data such as audio, video, or graphics, similar attributes can be defined. However, the information contained in the data can be diverse and difficult to characterize within a single application [46]. Two primary problems must be solved in order to provide content-based retrieval for unstructured data of these types. First, a means of extracting information contained in the unstructured data (e.g., an image) is required, and second, this information must be appropriately modeled in order to support user queries for content and data models for storage.

The first problem can be approached by segmentation and feature extraction. Content of unstructured data such as imagery or sound is easily identified by human observation; however, few attributes lead to machine identification. For still and moving images, it is possible to apply segmentation to yield features such as shape, texture, color, relationships among objects, and events [5]. This process is very difficult to apply to complex scenes; however, it can be simplified with *site models* which characterize images *a priori* and allow detection of scene changes. The second problem of managing the diverse query types and data models is simplified by semantic nets.

In our prototype system we currently use manual feature extraction of well-defined attributes which fit into a fixed data schema. The process of scene identification is semi-automated by detection of scene transitions. A similar approach has been applied by Swanberg et al. [42, 41] and MacNeil [25]. In the former work, application domain-specific models are used to capture extracted information and to support content-based retrieval. In Steven's work in the ALT project at CMU [39], knowledge of content, structure, and use of the content is embedded in video objects. This approach is applied to the generation of video sequences using a rule-base and leads to the selection of seamless concatenation of video sequences from many small video shots.

3.3 Time-Dependent Data Support for Audio and Video

In this section we introduce and overview our temporal data models for supporting TAC functionality and show how these models can be applied to the specific motion picture database application.

3.3.1 Temporal Modeling with n-ary Relations

Few representation techniques specify appropriate data structures that support subsequent TAC operations on a database schema. Our approach is to use a mapping from a specification methodology to a database schema using the timed Petri net (TPN) and the relational database model [19, 22]. In this case, temporal intervals and relationships are described by a timeline representation in an unstructured format, or by a TPN in a structured format. Using a TPN, temporal hierarchy can be imparted to the conceptual schema as sets of intervals bound to a single temporal relation are identified and grouped. For a model as simple as a movie consisting of a sequence of scenes, we can directly create the conceptual temporal schema (Fig. 5). In a more general multimedia application this representation can be quite complex.

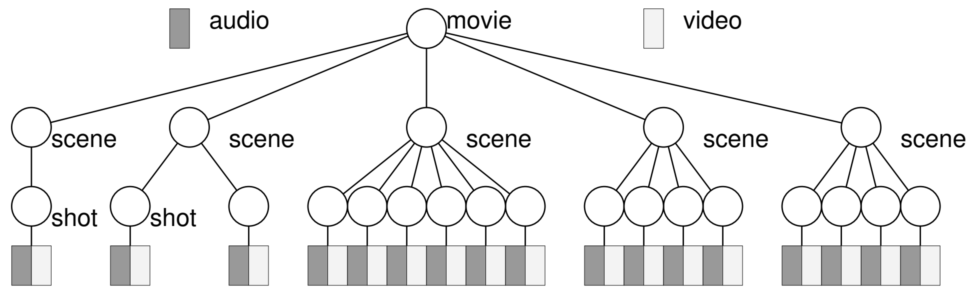


Figure 5: Temporal Schema for a Motion Picture

With this approach, the time-based representation is translated to a conceptual schema in the form of a temporal hierarchy representing the semantics of the original specification approach. Leaf elements in this model typically represent base multimedia objects (audio, video, text, etc.), but are only audio and video (and subtitles) for the VOD application. Timing information is also captured with node attributes, allowing the assembly of component elements during playout. Most proposed models for multimedia data are based on such a temporal-interval-based (TIB) representations (e.g., [10, 16, 17]), including our model [19, 22].

Temporal intervals can be used to model multimedia presentation by letting each interval represent the presentation or *playout* duration of some multimedia data element, such as a still image or an audio segment. The specification of timing for a set of intervals is achieved by indicating element durations for each data element and relative timing among elments. We have extended our initial TIB model to support n-ary temporal relations [19]. Like the binary case, the n-ary temporal relation is characterized by a start time (π_i), duration (τ_i), and end time for a data element i. The relative positioning and time dependencies are captured by a delay (τ_δ^i), as is the overall duration of a set of temporally related elements (τ_{TR}). A set of constraints exist for the n-ary temporal relations that can be used to determine whether an n-ary temporal relation is parallel or sequential, or if the temporal relation is known, verify its consistency in terms of temporal parameters.

This TIB modeling scheme can represent an arbitrary grain of interval (e.g., individual video frames or entire movies). However, the practicality of representing each frame by its temporal parameters is limited for audio and video data. Therefore, we group logically related frames into sets called *shots*, that are not normally decomposed. This aggregation corresponds to Fig. 5, and yields a blocking of sequences of continuous media as shown in Fig. 6. Within a block, data are assumed to have a homogeneous temporal characteristic (i.e, period of playout).

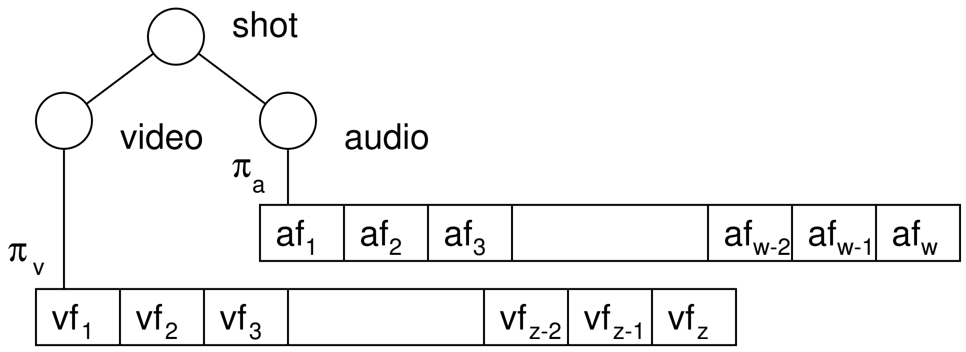


Figure 6: Blocking for Continuous Media

3.3.2 Physical Data Storage

Once a conceptual temporal model is established for a multimedia object, the multimedia data must be mapped to the physical system to facilitate database access and retrieval. For time-dependent multimedia data this presents some interesting challenges. Problems arise due to the strict timing requirements for playout of time-dependent data. For example, Fig. 7 shows the timing relationships within and between two streams of audio and video that must be preserved during playout.

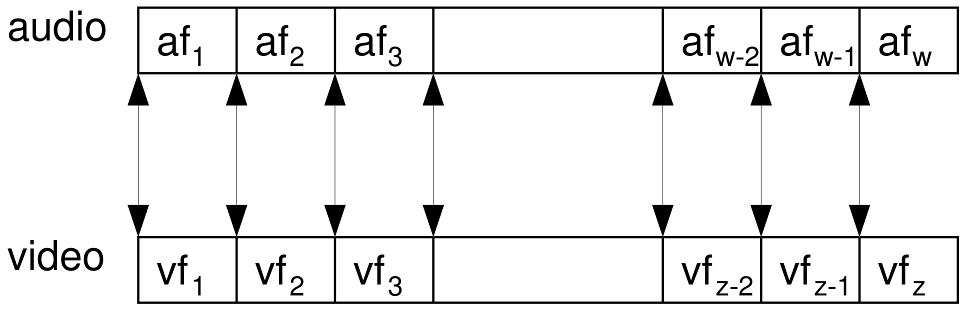


Figure 7: Timing for Audio and Video

From our TIB model it is possible to derive a playout schedule appropriate for the available resources at the time of delivery. This process is achieved by converting the relative timing specification of the TIB model to an absolute one, and scheduling this specification on the available resources [20]. Furthermore, it is possible to derive an absolute schedule from the TIB represtation in either the forward or reverse direction, or for only a fraction of the overall playout duration [19].

Satisfaction of the absolute timing specification is dependent on available resources. Intelligent layout of data on physical storage devices can significantly improve performance of this process. Approaches for the placement of audio and video data streams on a rotating-disk storage device have been investigated by Yu et al. [44, 47], Gemmell and Christodoulakis [15], and Rangan and Vin [30].

In addition to intelligent data organization, suitable operating system behavior is required for multimedia data delivery. Timing relationships between audio and video are preserved in the process of multimedia synchronization. Furthermore, in an application supporting audio and video playout such as our VOD prototype, nontraditional DBMS data access mechanisms are required to support TAC functionality. These TAC operations include *reverse, fast-forward, fast-backward, midpoint suspension, midpoint resumption, random access, looping*, and *browsing.* These operations are feasible with existing technologies, however, when non-sequential storage, data compression, data distribution, and random communication delays are introduced, the provision of these capabilities can be very difficult. Examples include viewing a motion picture backwards or reversing an animation of a series of images, rapid viewing of a long sequence of images (fast-forward or fast-backward), and stopping and starting of a motion picture (midpoint suspension and resumption).

Extensive work has now been published on system support for multimedia. Representative of this work is that of Anderson et al. in the Acme continuous media I/O server [1], and Rangan et al. [29, 43] for storage organization and access control to support multiple sessions from storage and communications devices.

4 The Virtual Video Browser System

The Virtual Video Browser is a true VOD application that permits an individual to browse an electronic video collection stored in a digital VOD database. The basic objective in the development of this prototype was to illustrate the capabilities of our temporal modeling scheme and to gain experience with an interesting multimedia application that requires the convergence of many technologies. Secondary objectives were to design a software system with various desirable properties including *ease of use and understanding*; and *ease of maintainability, portability, and extensibility.*

The VVB operates in association with an extensive multimedia database. Movies and scenes of movies can be identified and viewed through the VVB based on movie-specific attributes including actor names, director names, and scene characteristics. Movie and scene content indexing are facilitated by summary, keyword, and transcript searching. In addition to viewing scenes from the movies, the user can view all of a movie's textual information including summaries and transcripts.

In the remainder of this section we describe the details of the VVB operation, database, and software architecture.

4.1 VVB Operation

The VVB can be characterized by five operational modes and associated interfaces. These correspond to the (1) opening menu, (2) virtual video shelves, (3) query input, (4) query output, and (5) video playout interfaces.

The opening menu of the VVB displays images representing each of the available categories. The user can choose a category by the click of a mouse button. The each genre is identified by a picture. For example, the “western” category is indicated by a desert scene with cacti. Once a category is selected, the *virtual shelf screen* appears, showing an iconic representation of shelves of a video rental store.

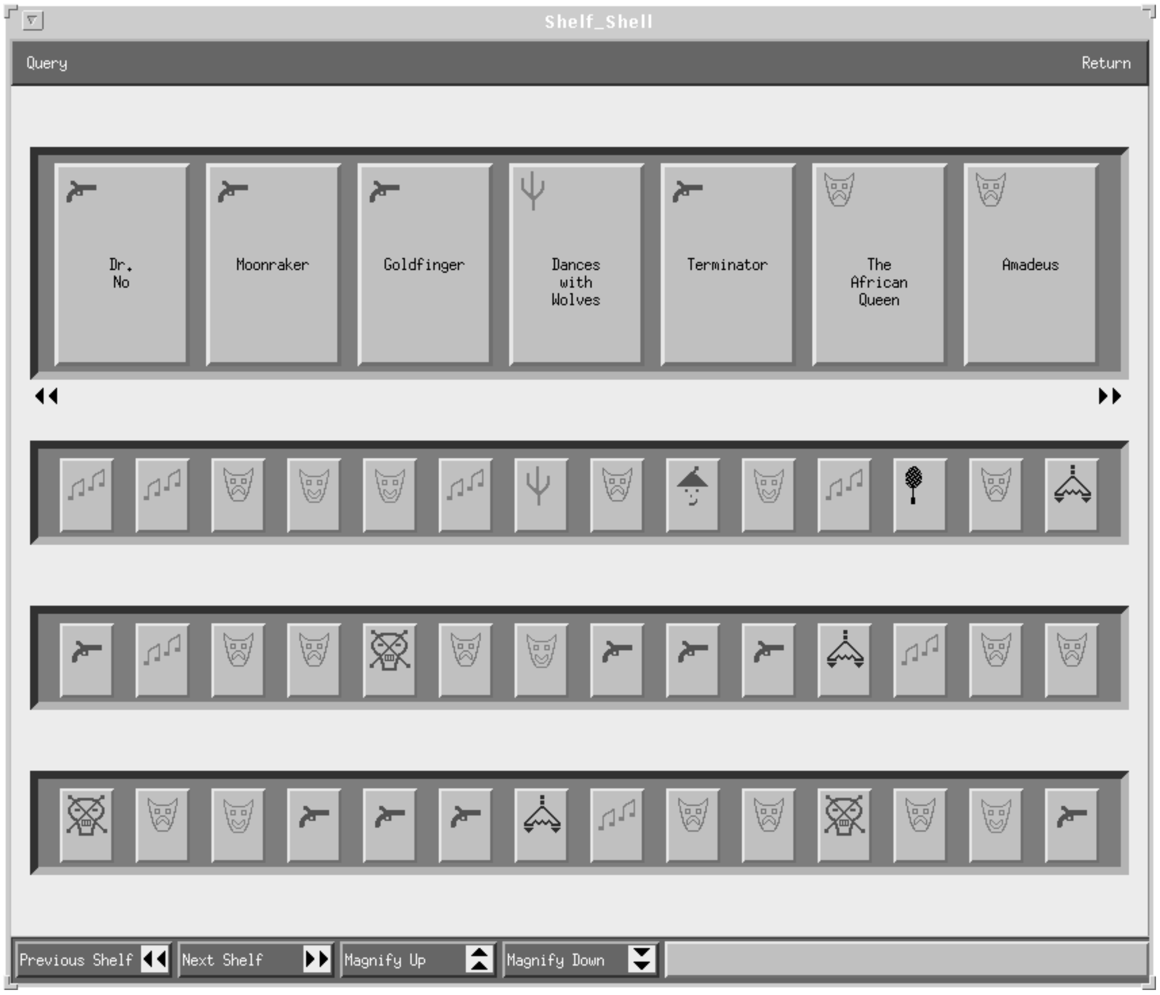


Figure 8: Virtual Shelf Screen

The *virtual shelf screen* is a virtual display of the available movie cassettes as patterned after a video rental store.[3] Each of the movies on the virtual shelf screen is represented by a rectangular icon which resembles the shape of an actual VHS video box. To help distinguish movies of different categories on the video shelf screen, each of the movies on the screen contains an icon which represents one of the available movie categories. If a large number of videos are to be displayed, then the user has the ability to move between multiple shelves with the use of the arrows on the bottom of the screen.

[3]Loeb [24] mimics a jukebox in a similar manner to achieve rapid user acceptance.

The virtual shelf screen is exited when a movie icon or query input screen is selected. Selection of a movie icon brings up the query output and video playout screens. When the query button is pressed, the *query input screen* appears.

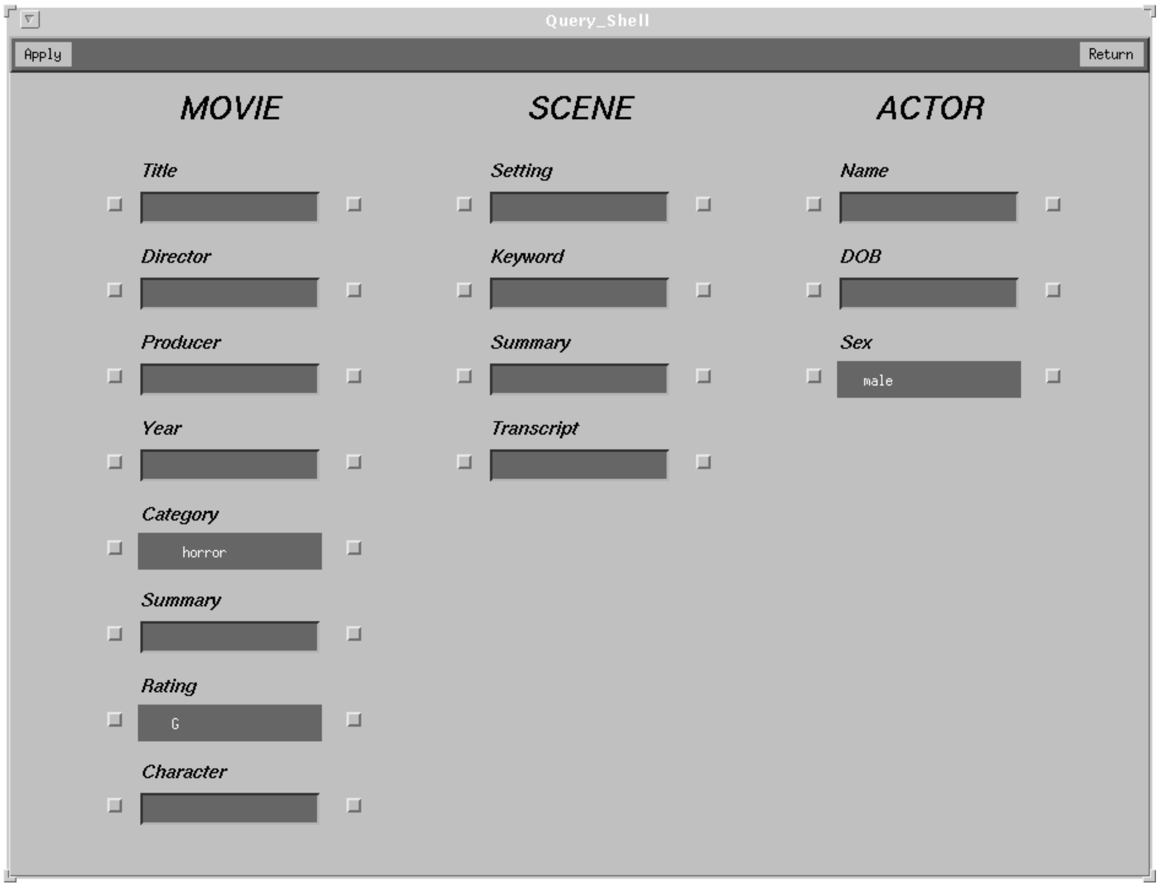
The *query input screen* allows the user to create database queries in the identification of scenes for viewing or to identify a desired movie. This interface permits the user to make queries to the database on any content within the realm of movies, scenes, or actors (Fig. 9). The text input boxes for each attribute of this screen contain two toggle buttons. The left toggle button is depressed to indicate consideration of an input attribute (relational database *selection*). For example, if the user wishes to query on all movies directed by Oliver Stone, the name "Oliver Stone" is typed into the text box for director and the toggle button to the left is depressed. If the user wishes to query for all movies directed by Oliver Stone in the year 1987, then the user enters "Oliver Stone" in the director text box and "1987" in the year text box. The toggle buttons to the left of director and year must then be depressed to apply these as inputs attributes. Similarly, the right toggle buttons choose the desired information returned from the query (relational database *projection*).

In addition to the iconic graphical representation of the selected movies, the VVB can display any textual data from the database by using available attributes of title, director, actors, year, category, synopsis, etc. This functionality supports quick scanning of the database to find a movie of interest. The *query output screen* is invoked by performing database queries using the database query input screen. Each time the "Apply" button is depressed on the database query screen, a query is issued to the database manager and a list of movies is returned. By only choosing a subset of the available output attributes, this window automatically is sized to the returned data fields.

The set of input and output application-specific relations accessible for query are *movie, scene*, and *actor*. Their attributes are summarized in Table 1.

Given the above attributes the following example database queries are possible:

- Find the scene in which Humphrey Bogart says "Play it again Sam."
- Find all actors present in the opening scene of all remakes of *Dracula.*
- Find the actors who have been directed by Oliver Stone.
- Find the Cafe scenes in *Casablanca.*
- Find all of the horror movies created in 1989.



Category: horror

Rating: G

Sex: male

Figure 9: Query Input Screen

Table 1: Motion Picture Database Attributes

Movie (M)	Scene (S)	Actor (A)
title	keywords	name
director	setting	dat of birth
producer	summary*	sex
year	transcript*	
category		
summary*		
character		

- Find all movies directed by Albert Broccoli in 1962 which were dramas.

Once a movie or scene is selected with the mouse from the virtual shelf or query output screens, the *video playout screen* appears. This screen provides the visual interface for displaying different parts of a movie and is supported by TAC buttons. A user is permitted to scan from scene to scene beginning with the initial selection.

4.2 The VVB Database

The VVB database is implemented using the POSTGRES DBMS [37, 45] and several data schemata. These schemata provide application-specific data access based on content as well as TAC functionality through our temporal models. A third data schema relates the application-specific data model to the temporal model. These schemata and their relationships are illustrated in Fig. 10 using a network representation.[4]

4.2.1 Temporal Schema

The temporal schema captures the relative timing relationships between components of the movies as described in Section 3.

4.2.2 Application-Specific Schema

The application-specific schema defines the VVB system application database to which queries can be made. It contains six components, implemented as relations: movie, scene, actor, movie-scene, scene-actor, and movie-actor. The actor relation describes actors and actresses. The movie-scene relation defines the parent-child relation which identifies all movies and their scenes. The scene-actor relation defines the relation to identify which actors are in which scenes. The movie-actor relation defines the relation to identify which actors are in which movies. Example data in these fields are illustrated in Tables 2–5.

4.2.3 Schema Interface

In order to relate the application-specific schema for a movie database to the temporal schema we use the schema interface. For the VVB application, this component consists of

[4]This structure is implemented using relations in POSTGRES.

Table 2: Example Data Fields: Actor

actor_id	name	date of birth	sex
001	Sean Connery	August 25, 1930	male
002	Ursula Andress	March 19, 1936	female

Table 3: Example Data Fields: Movie_Actor

movie_id	actor_id	character
001	001	James Bond
001	002	Honeychile Ryder

Table 4: Example Data Fields: Movie

movie_id	title	director	producer	year	category	summary	liner
001	Dr. No	Young	Broccoli	1962	action	sum1	img1

Table 5: Example Data Fields: Scene

scene_id	keywords	setting	summary	transcript
001	license to kill	Crab Key Island	Bond swimming	Oh James
0010	skiing	Swiss chalet	Bond skiing	Shall we ski?...

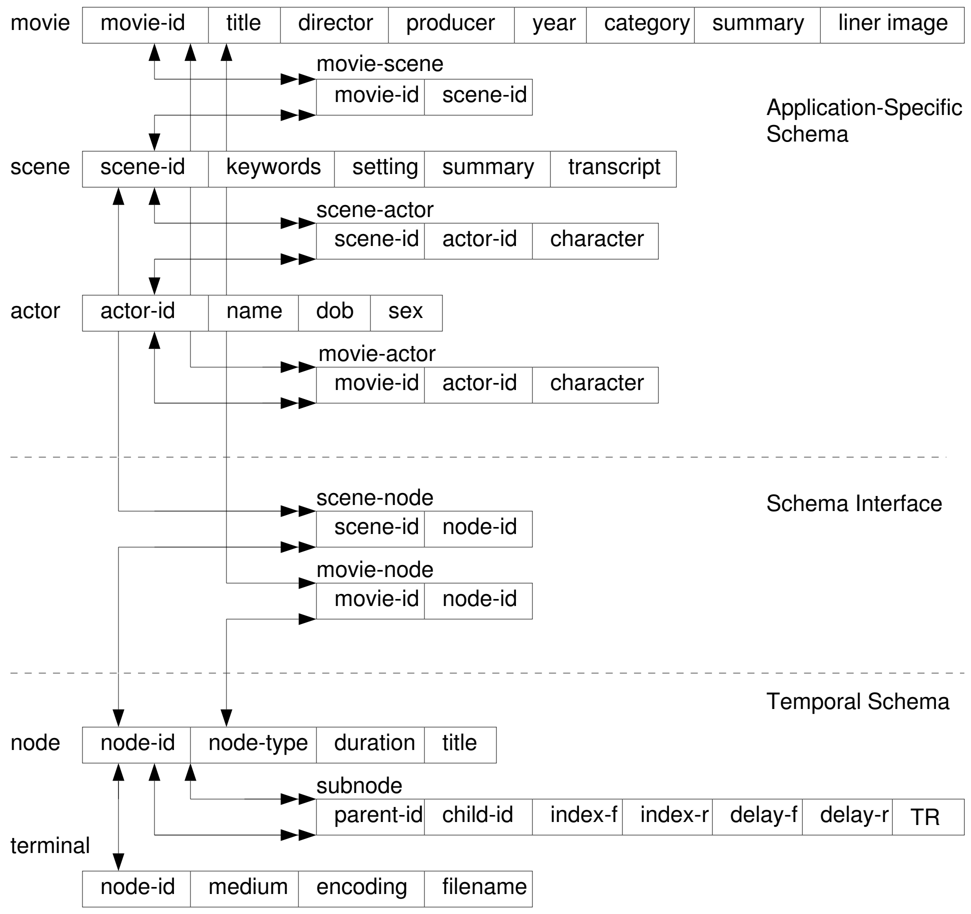


Figure 10: Database Schema for VVB System

two relations, scene-node and movie-node, as defined in Fig. 10.

4.2.4 Query Engine

The query engine for the VVB records the input selection from the query input interface in a data structure called DBQin. For each selected input attribute, a corresponding component is generated in the PQUEL query. Similarly, each selected output field results in a term in the PQUEL query. The make_query subprogram then formulates the PQUEL text string as:

retrieve <determined by output selections >
from < m in movie, s in scene, a in actor>
where <determined by input selections>

The text string is built by retrieving from tables the appropriate strings and adding the user input text as appropriate. This final PQUEL string is passed to the POSTGRES DBMS where it is executed, resulting in the return of the selected output attributes.

4.3 Software Architecture

In order to meet our objectives of application flexibility, extensibility, portability, and maintainability, we have designed the VVB system software in a layered fashion. This simplifies the replacement of software components with changing technology. For example, we can easily replace the video compression/decompression subsystem to accommodate new algorithms and hardware improvements.

The software design of the VVB system is divided into three distinct components [12]. These components are the user interface, the database, and the imaging system. An application programming interface (API) has been created to provide functionality in each case. The actual VVB application is implemented on top of the APIs to satisfy the VVB requirements. A particular advantage with this design methodology is the ability to create new applications from existing code because most of the VVB functionality is implemented within APIs. Below is a diagram of the three different APIs and how they relate to the commercial off-the-shelf (COTS) products and the VVB application (Fig. 11).

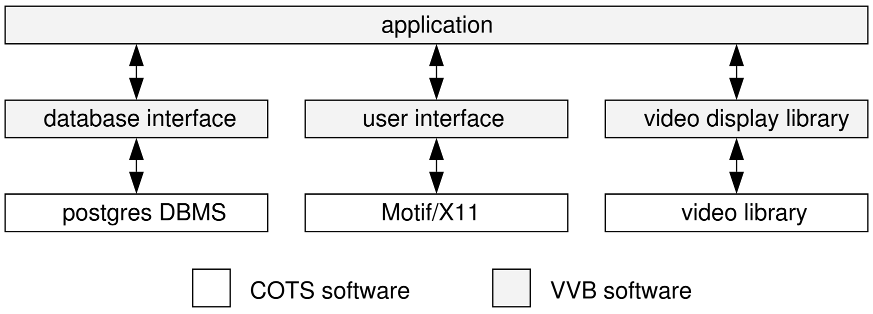


Figure 11: Software Architecture of the VVB System

The VVB system is implemented in C on a Unix platform and uses X11, Motif and POSTGRES. Digital video recording and playback are facilitated by XVideo hardware from Parallax Graphics Inc.

5 Discussion

The VVB was developed primarily as a demonstrative application for our previous work in the modeling of time-dependent data. To this end it has satisfactorily achieved its purpose (i.e., we are able to specify and store temporal relationships and use them in media playout). However, the inherent sequential characteristics of conventional movies limit the application of our temporal models. For this reason we are pursuing additional applications which require the full range of TAC functionality. The VVB also illustrates our interest in investigating content-based retrieval from video databases.

The VVB can be evaluated from several other perspectives. In particular, these are (1) number of supported users, (2) database retrieval performance, (3) ease of use, (4) diversity of semantic database content (5) complexity of database maintenance.

In our current configuration, the number of supported users is limited by the availability of compatible video decompression hardware. With additional workstations served by the same video database the performance bottleneck becomes the available bandwidth of the communication medium (Ethernet in our testbed) and the I/O performance of the database server. Selection of number and performance of storage and communications components in a VOD scale-up is dependent on component costs and degree of supported VOD functionality. For example, our studies indicate the feasibility of supporting up to five true-VOD sessions (users) from a conventional hard disk storage device [6]. Performance limitations are due primarily to component bandwidth limitations and not the behavior of the VVB interface.

The VVB has been successful in attaining our objective of ease of use. This quality, necessary for the VVB to gain user acceptance, was achieved through the development of a simple human-computer interface using fast prototyping and numerous iterations with user feedback.

In terms of diversity of supported semantic content, the VVB is limited by our choice of domain-specific attributes. In order to support a wide range of queries (e.g., “find a scene containing a dog”), attributes are required to be identified *a priori*, or supported dynamically. The former requires knowledge of a large set of potential queries. A dynamic approach would search for content in available data including in the transcripts, audio tracks and video segments. The VVB can find the scene with the dog assuming that “dog” is identified in the summary/transcript for the appropriate scene. However, a more interesting solution would interpret a diverse set of input queries and apply them on the unstructured audio and video data as well as to an established semantic net containing pre-extracted information. This requires image segmentation and feature extraction which are not implemented in the VVB.

Related to dynamic attribute identification is database maintenance. As movies are created and added to the VVB database, movie content must be extracted and indexed. This process is not currently automated. However, we are currently developing a semi-automated procedure for parsing movies using image and audio processing. As a basis, we have successfully used video dynamics as indicated by frame size to identify scene transitions.

6 Conclusion

In this paper we have overviewed the various technologies required to support a content-based retrieval from a multimedia information system encompassing video data. We have also described the specific application instance of video-on-demand, and described our prototype VOD system for digital motion pictures. The system demonstrates our conceptual modeling scheme for time-dependent multimedia data and serves as a basis for a more general system supporting diverse multimedia data types.

In the future, we plan on extending our system to support distributed databases, semantic models for information capture, automatic feature extraction/segmentation, and additional applications including on-demand course instruction.

References

[1] Anderson, D.P., Homsy, G., "A Continuous Media I/O Server and Its Synchronization Mechanism," *Computer*, Vol. 24, No. 10, October 1991, pp. 51-57.

[2] Andrews, E.L., "Cable Concern Plans to Offer 500 Channels," *New York Times*, December 2, 1992.

[3] Bender, W., Lie, H., Orwant, J., Teodosio, L., Abramson, N., "Newspace: Mass Media and Personal Computing," *Proc. Summer 1991 Usenix Conf.*, Nashville, Tennessee, June 1991, pp. 329-348.

[4] Cambell, A., G. Coulson, F. Garcia and D. Hutchison, "A Continuous Media Transport and Orchestration Service," *SIGCOMM'92* Baltimore, Maryland, August 1992.

[5] Carlotto, M.J., "Image Understanding Now and in the Future," *Advanced Imaging*, Vol. 7, No. 8, August 1992, p. 26.

[6] Chen, H.-J. and T.D.C. Little, "A File System for Multimedia Applications," Tech. Rept. 12-09-1992, Multimedia Communication Laboratory, Boston University, 1992.

[7] Christodoulakis, S., Faloutsos, C., "Design and Performance Considerations for an Optical Disk-based, Multimedia Object Server," *Computer*, December 1986, pp. 45-56.

[8] Committee Draft International Standard, "Information Technology Hypermedia/Time-based Structuring Language (HyTime)," ISO/IEC CD 10743, April 1, 1991.

[9] Dimitrova, H., Golshani, F., "EVA: A Query Language for Multimedia Information Systems," *Proc. 1992 Workshop on Multimedia Information Systems*, Tempe, Arizona, February, 1992, pp. 1-20.

[10] Committee Draft International Standard, "Information Technology – Standard Music Description Language (SMDL)," ISO/IEC CD 10743, April 1, 1991.

[11] Davenport, G., T.G.A Smith, and N. Pincever, "Cinematic Primitives for Multimedia," *IEEE Computer Graphics & Applications*, July 1991, pp. 67-74.

[12] Folz, R.J., F.W. Reeve, D.H. Schelleng, "Virtual Video Browse Software Documentation," Multimedia Communications Laboratory, Boston University, 1992.

[13] Gelman, A.D., H. Kobrinski, L.S., Smoot, S.B., Weinstein, "A Store-and-Forward Architecture for Video-On-Demand Service," *Proc. ICC*, 1991, pp. 27.3.1-27.3.5.

[14] Gelman, A.D., Halfin, S., "Analysis of Resource Sharing in Information Providing Services," *IEEE Global Telecommunications Conference and Exhibition (Globecom '90) Conference Record*, San Diego, CA, December 1990, pp. 312-316.

[15] Gemmell J., and S. Christodoulakis, "Principles of Delay-Sensitive Multimedia Data Storage and Retrieval," *ACM Trans on Information Systems*, Vol. 10, No. 1, January 1992, pp. 51-90.

[16] Herrtwich, R.G., "Time Capsules: An Abstraction for Access to Continuous-Media Data," *Proc. 11th Real-Time Systems Symp.*, Lake Buena Vista, FL, December 1990, pp. 11-20.

[17] Herrtwich, R.G., Delgrossi, L., "ODA-Based Data Modeling in Multimedia Systems," *International Computer Science Institute Tech. Rept.* TR-90-043, August 1990.

[18] Lippman, A., Bender, W., "News and Movies in the 50 Megabit Living Room," *Globecom '87*, Tokyo, Japan, November 1987, pp. 1976-1981.

[19] Little, T.D.C., and A. Ghafoor, "Interval-Based Temporal Models for Time-Dependent Multimedia Data," to appear in *IEEE Transactions on Data and Knowledge Engineering*, August 1993.

[20] Little, T.D.C., Ghafoor, A., "Scheduling of Bandwidth-Constrained Multimedia Traffic," *Computer Communications*, Vol. 15, No. 6, July/August, 1992, pp. 381-387.

[21] Little, T.D.C., and A. Ghafoor,"Spatio-Temporal Composition of Distributed Multimedia Objects for Value-Added Networks," *Computer*, Vol. 24, No. 10, October 1991, pp. 42-50.

[22] Little, T.D.C., Ghafoor, A., "Synchronization and Storage Models for Multimedia Objects," *IEEE J. on Selected Areas in Comm.*, April 1990, pp. 413-427.

[23] Loeb, S., "Delivering Interactive Multimedia Documents over Networks," *IEEE Communications*, Vol. 30, No. 5, May 1992, pp. 52-59.

[24] Loeb, S., Hill., R., Brinck, T., "Lessons from LyricTime: A Prototype Multimedia System," *Proc. 4th IEEE ComSoc Intl. Workshop on Multimedia Communications (Multimedia '92)*, Monterey, CA, April 1992, pp. 106-113.

[25] MacNeil, R., "Generating Multimedia Presentations Automatically using TYRO, the Constraint, Case-Based Designer's Apprentice," *Proc. 1991 IEEE Workshop on Visual Languages*, Kobe, Japan, October 1991, pp. 74-79.

[26] O'Docherty, M.H., Daskalakis, C.N., "Multimedia Information Systems - The Management and Semantic Retrieval of All Electric Data Types," *The Computer Journal*, Vol. 34, No. 3, June 1991, pp. 225-238.

[27] Ramarao, R., and V. Ramamoorthy, "Architectural Design of On-Demand Video Delivery Systems: The Spatio-Temporal Storage Allocation Problem," *Proc. ICC'91*, 1991, pp. 17.6.1-17.6.5.

[28] Rangan, P.V., and H.M. Vin, "Efficient Storage Techniques for Digital Continuous Multimedia," to appear in *IEEE Transaction on Knowledge and Data Engineering*, August 1993.

[29] Rangan, P.V., H.M. Vin, and S. Ramanathan, "Designing an On-Demand Multimedia Service," *IEEE Communications Magazine*, Vol. 30, No. 7, July 1992, pp. 56-64.

[30] Rangan, P.V., Vin, H.M., "Designing File Systems for Digital Video and Audio," *Proc. 13th Symp. on Operating Systems Principles (SOSP'91), Operating Systems Review*, Vol 25, No. 5, October 1991, pp. 81-94.

[31] Rowe, L.A., B.C. Smith, "A Continuous Media Player," *Proc. 3rd International Workshop on Network and Operating System Support for Digital Audio and Video*, La Jolla, CA, November 1992.

[32] Sasnett, R.M., "Reconfigurable Video," *M.S. Thesis*, MIT, 1986.

[33] Silvers, Singh, E., "Multimedia Communications on the NYNEX Shuttle," *Proc. COMPCON Spring 1992*, San Francisco, CA, February 1992, pp. 84-87.

[34] Sincoskie, W.D., "System Architecture for a Large Scale Video On Demand Service," *Computer Networks and ISDN Systems* Vol. 22, 1991, pp. 155-162.

[35] Aguierre Smith, T.G.A., and Davenport, G., "The Stratification System: A Design Environment for Random Access Video," *Proc. 3rd International Workshop on Network and Operating System Support for Digital Audio and Video*, La Jolla, CA, November 1992.

[36] Aguierre Smith, T.G., Pincever, N.C., "Parsing Movies in Context," *Proc. Summer 1991 Usenix Conf.*, Nashville, Tennessee, June 1991, pp. 157-168.

[37] Stonebraker, M., and G. Kemnitz, "The POSTGRES Next Generation Database Management System," *Communications of the ACM*, Vol. 34, No. 10, October 1991, pp. 78-92.

[38] Stotts, P.D., and R. Furuta, "Petri-Net-Based Hypertext: Document Structure with Browsing Semantics," *ACM Trans. on Office Automation Systems*, Vol. 7, No. 1, Jan. 1989, pp. 3-29.

[39] Stevens, S.M., "Embedding Knowledge in Continuous Time Media," *Proc. 2st Intl. Workshop on Network and Operating Support for Digital Audio and Video*, Heidelberg, Germany, November 1991.

[40] Sutherland, J., and L. Litteral, "Residential Video Services," *IEEE Communications Magazine*, Vol. 30, No. 7, July 1992, pp. 36-41.

[41] Swanberg, D., C.F. Shu, and R. Jain, "Architecture of a Multimedia Information System For Content-Based Retrieval," *Proc. 3rd International Workshop on Network and Operating System Support for Digital Audio and Video*, La Jolla, CA, November 1992.

[42] Swanberg, D., Weymouth, T., Jain, R., "Domain Information Model: An Extended Data Model for Insertions and Query," *Proc. 1992 Workshop on Multimedia Information Systems*, Tempe, Arizona, February, 1992, pp. 39-51.

[43] Vin, H.M., and P.V. Rangan, "Designing a Multi-User HDTV Storage Server," *IEEE Journal on Selected Areas in Communications*, January 1993.

[44] Wells, J., Yang, Q., Yu, C., "Placement of Audio Data on Optical Disks," *Proc. 1st Intl. Conf. on Multimedia Information Systems '91*, Singapore, January 1991, pp. 123-134.

[45] Wensel, S., Ed., "The POSTGRES Reference Manual," Report M88/20, Electronics Research Laboratory, University of California, Berkeley, CA, March 1988.

[46] Wittenburg, T.M., "A Reconfigurable Multimedia Document Management System Based on an Extended Object-Oriented Data Model," *M.S. Thesis*, Dept. of Electrical, Systems and Computer Engineering, Boston University, December 1992.

[47] Yu, C., Sun, W., Bitton, D., Yang, Q., Bruno, R., Tullis, J., "Efficient Placement of Audio Data on Optical Disks for Real-Time Applications," *Comm. of the ACM*, Vol. 32, No. 7, July 1989, pp. 862-871.