# Distributed Multimedia Digital Libraries for Medical Education

## Introduction

Recent advances in multimedia database technology and high performance networking present an extraordinary opportunity to transform basic and continuing medical education by bringing search, discovery and presentation capabilities for multimedia information directly into medical classrooms. Vast collections of medical videos, clinical slides and laboratory images which hold enormous value for medical faculty and physicians remain a largely unused resource because the information is neither digitized nor cataloged and, as a result, not easily accessed or used. The system developed through our research will enable content-based query and retrieval for digital medical libraries, supporting basic medical education courses in medical schools and offering continuing medical education to physicians in rural, medically underserved areas. The proposed research will investigate web-based content-aware methods for locating, analyzing and presenting video, image, text, and audio data stored across distributed databases.

The key objectives of the proposed project are

- Create a large, permanent, indexed multimedia medical archive as a relevant and effective resource for medical education.
- Extend the functionality of underlying multimedia database technology to advance the accessibility and usability of multimedia databases, including complex multi-feature query processing, advanced stream handling, and extended storage hierarchies.
- Expand the capabilities of our prototype medical education query and retrieval system to incorporate agent-based components for intelligent processing of high-level user "concept" queries and secure data access of distributed multimedia medical data.

In cooperation with external partners from medical school faculty and the health care industry, the advanced prototype medical education system and digital medical archive developed through the proposed research will be deployed, evaluated and refined.

## Motivation and Contribution

The need for content-based query and retrieval for digital multimedia medical archives has been documented by regional medical centers serving rural communities and by medical school faculty. Medical centers that encompass rural areas are interested in developing telemedicine networks that provide improved connectivity and medical education options for physicians, in particular in underserved, "critical access" areas.  These physicians need additional support in the form of training and collaboration. Medical centers are investigating innovative approaches for delivering support through high quality web-based models, resulting in improved stewardship of resources and lower cost education options. Essential to their mission is the development of models based on digital and communications technology, where large parts of the medical curriculum are presented in a format that utilizes digital multimedia data. A key requirement is support for content-based query and retrieval of multimedia medical data addressing critical areas in continuing medical education. In addition, a strong base of medical faculty is committed to the development of prototype multimedia medical education systems for the basic medical curriculum that would deliver unparalleled medical education to medical students. Their concept for a medical education prototype incorporates sophisticated multimedia class notes, sample case-based teaching materials, and access to a large, digital, indexed multimedia medical archive. The archive would be used for search and discovery by medical faculty to support interactive preparation and delivery of lectures, assignments and exams using content-based query, retrieval and presentation of multimedia medical data. In particular, the system should support online classroom query, retrieval and streaming for medical video clips.

Several national initiatives have been moving in this direction. The Digital Libraries Initiative supports the creation of large, accessible knowledge bases and the development of methods to improve their usability. The spectrum of knowledge bases created through this program includes digital collections for history, music, art and news. Some collections support medical information retrieval, but these do not investigate the issues of image feature extraction, video query operators and media streaming that are required for video content-processing, search and presentation. The Informedia Project does provide some image-processing for their digital news collection, but their search and summarization techniques focus on audio-processing in conjunction with speech recognition and natural language processing. The National Library of Medicine is solely concerned with medical data and provides high-quality systems for comprehensive and convenient access to medical information. Their databases and information files, however, have not fully addressed the video component and do not provide significant support for image feature processing, video indexing, image queries, stream processing, and multimedia presentation.

The ultimate objective of our proposed research and development is to establish and advance the web-based delivery of distributed multimedia medical data as an integral part of medical education. We plan to create a large, digital archive of multimedia medical data that is relevant to medical education, with particular focus on video and image. The archive management system will provide content-based search and discovery through multimodal user queries, and will support the efficient retrieval and presentation of video, image, text, and audio information. We will investigate and develop components that are needed to support the accessibility and usability of distributed digital medical archives. This encompasses two principal areas of research: extension of the underlying multimedia database technology (Figure 1) and investigation into issues related to the development of agent-based components for intelligent processing of high-level queries and secure access of distributed multimedia data (Figure 2).

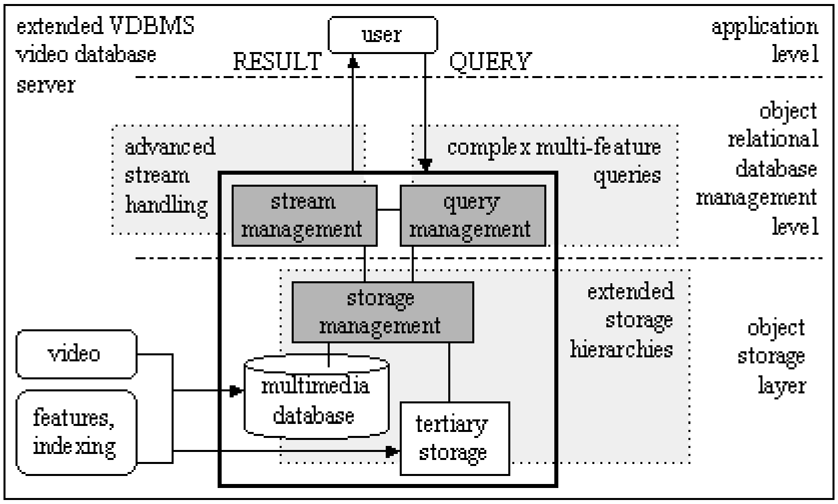


**Figure 1. Proposed research areas for advancing multimedia database technology.**

The proposed research will address the following critical issues:

1. *Complex multi-feature multimedia query operators*. To support image similarity search for complex multi-feature queries, we will introduce new query operators that handle optimal aggregate ranking for any number and combination of image features, ensuring that the operators are practical for real-world databases and can easily be integrated into query pipelines to support a hierarchy of join operations, nested views and a wider range of query execution plans.
2. *Advanced stream handling*. Efficient presentation of media streams in a search and discovery database environment requires stream management policies that consider the inherent connection between query results and user stream requests. We propose a new policy that

processes “expected streams” to reduce initial latency and disk I/O. We also plan to address the query processing of stream data. This is a new and growing research area, in particular in the case of joining data streams to answer user queries. We will develop query operators that are capable of joining multiple infinite data streams and combined streamed/stored data input. We also plan to address access control of streaming video, where video data is altered to protect privacy or ownership while streaming from the database to the user.

3. *Extended storage hierarchies*. New database components are needed to handle extended storage hierarchies that support real-time access to buffer, disk and tertiary storage. The large-scale storage for video data requires efficient access and management of tertiary resident data. We propose methods that make this data directly accessible to the system, including optimal scheduling algorithms and caching schemes. We also intend to consider the Internet as a source of multimedia data.
4. *Expanded capability for our prototype medical education system.* A system that delivers unified access to distributed multimedia data requires the development of a number of related web-based components. We will focus on building agent-based components that support 1) processing for high-level user queries directed to a collection of heterogeneous systems containing a combination of multiple data types, 2) secure data access for the sensitive shared medical information, and 3) processing of retrieved multimedia results for presentation.

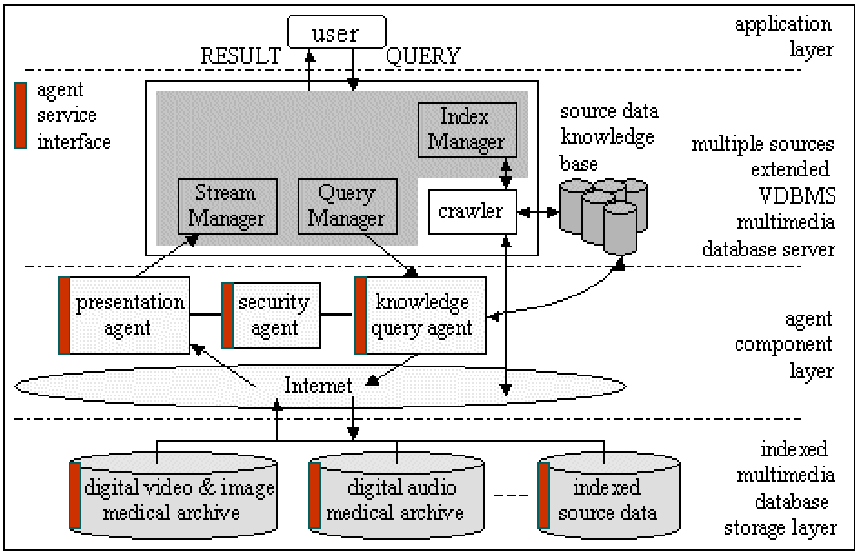


**Figure 2. Agent-based components supporting unified, secure access to distributed multimedia databases.**

We will collaborate with our partners in health care and medical education to acquire several hundred hours of high-quality medical education videos and to establish classroom scenarios and faculty/physician requirements for creating the advanced medical education prototype system.

## Previous Related Research

A five-year multidisciplinary focus on research in multimedia database management systems at Purdue University has achieved significant results. The Video Data Base Management (VDBMS) research initiative has successfully developed a video-enhanced database system that supports comprehensive and efficient database management capabilities for digital video libraries. Our fundamental concept was to provide a full range of functionality for video as a fundamental database object, with its own description, parameters and methods. The diagram in Figure 3 illustrates our layered system architecture, consisting of the object storage system layer at the bottom, the object relational database management layer in the middle, and the user interface layer at the top. VDBMS provides complete video database management, including 1) video preprocessing for feature and content analysis to produce meta-data descriptors which represent video content, 2) video and meta-data storage management, 3) image- and content-based query processing, 4) buffer and storage management, and 5) continuous media streaming. The system has been developed as a research platform and supports the entire process of investigating, implementing, analyzing and evaluating new techniques, thus identifying in a concrete way which techniques are truly practical and robust. The framework is flexible and extensible; new algorithms, new data types, or entirely new components can easily be added since system components have been designed with well-defined interfaces and encapsulated functionality. Our system has been used for testing component performance and for testing the impact of new components on existing ones. The Query, Stream, Buffer and Storage Managers in the current VDBMS system will be extended and adapted to address the advanced multimedia database functionality described in this proposal. We now briefly discuss the significant advances resulting from our current multimedia database initiative, and include results from the related Multimedia Support Infrastructure (MSI) project[1] and from the Indiana Telemedicine Incubator (ITI) project[2].

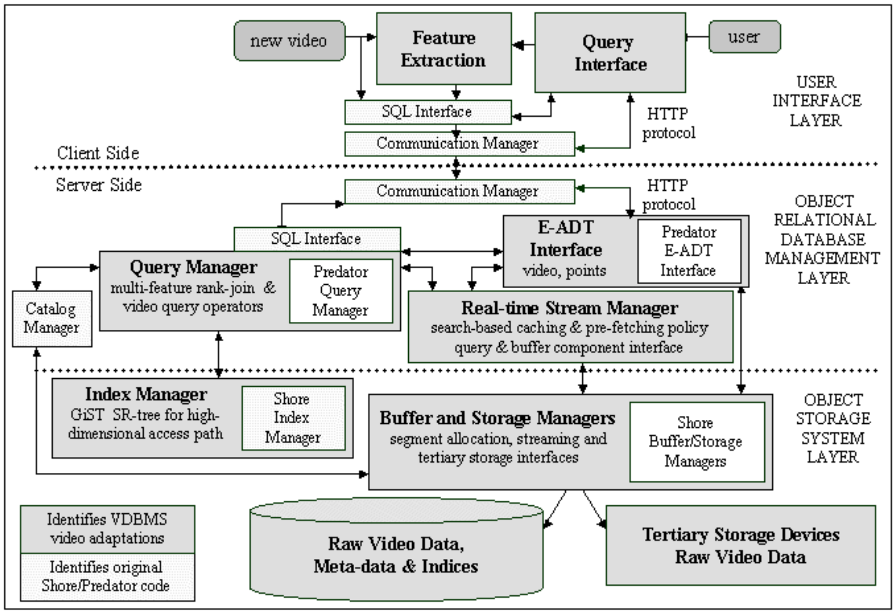

new video
Feature Extraction
Query Interface
user
SQL Interface
HTTP protocol
USER INTERFACE LAYER
Communication Manager
Client Side
Server Side
Communication Manager
HTTP protocol
OBJECT RELATIONAL DATABASE MANAGEMENT LAYER
SQL Interface
E-ADT Interface
video, points
Predator E-ADT Interface
Query Manager
multi-feature rank-join & video query operators
Predator Query Manager
Catalog Manager
Real-time Stream Manager
search-based caching & pre-fetching policy
query & buffer component interface
Index Manager
GiST SR-tree for high-dimensional access path
Shore Index Manager
Buffer and Storage Managers
segment allocation, streaming and tertiary storage interfaces
Shore Buffer/Storage Managers
OBJECT STORAGE SYSTEM LAYER
Identifies VDBMS video adaptations
Identifies original Shore/Predator code
Raw Video Data, Meta-data & Indices
Tertiary Storage Devices
Raw Video Data

**Figure 3. Current architecture for the VDBMS multimedia database management system, with targeted extensions identified for the Query, Stream, Buffer, and Storage Manager components.**

## Research Advances in Multimedia Databases.

Digital video is a complex data type, requiring a significant amount of preprocessing to extract information that represents video content. The representative information is used to index the video for database query and retrieval. Video data is inherently large and its representative indexing information can be even larger, especially in the case of image-based feature indexing. Methods for handling traditional data storage, query and retrieval cannot be extended to provide this functionality for video. Furthermore, database management systems that either hide video data as a Binary Large OBject (BLOB) or store the video outside the database cannot support meaningful processing and optimization. The database cannot operate on video content when videos are stored as BLOBs, and the database has no control over the video data itself when videos are stored in flat files (as in Virage.) We have abstracted the video as a well-defined and fully functional data type. We store video data inside the database together with its content

[1] Elmagarmid A., Ghafoor, A., Park, K., Spafford, E. and Coppoc, G. MSI: *A* Research Infrastructure for Integrated Quality of Service Management in Multimedia Computing Environments, NSF EIA-9972883. 9/99-9/04, $1,386,612.

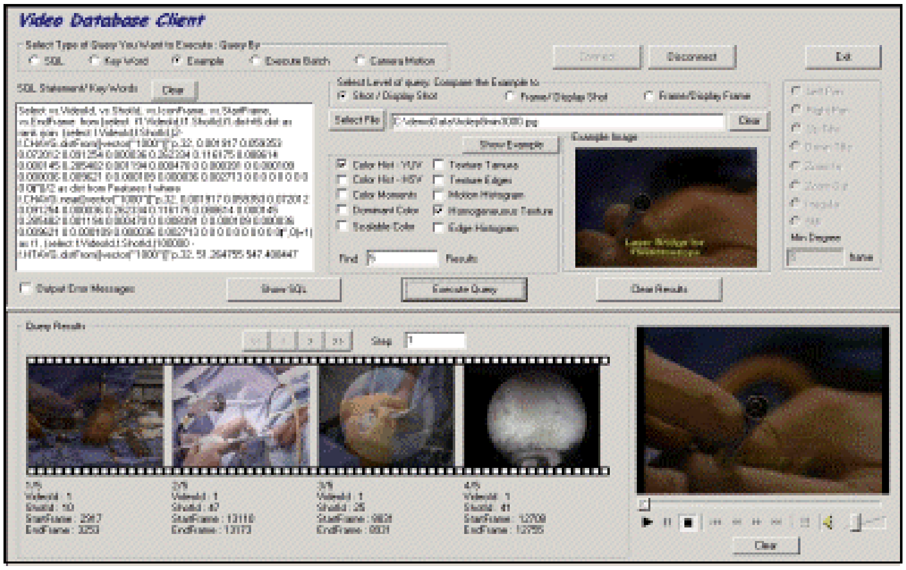
[2] Elmagarmid A., Aref, W., Ghafoor, A., Prabhakar, S., Coppoc, G. and Trippi, J. Indiana Telemedicine Incubator: A multidisciplinary consortium for the development of distributed multimedia database technology for the health care industry. State of Indiana 21st Century Research and Technology Fund. 6/00-6/02, $1,698,880.

representation, giving our system absolute control over every interaction with the video. This is essential if we want video database applications to inherit all the powerful functionality generally provided by traditional database management systems.

The VDBMS query model uses the features approach in searching video by content. VDBMS has developed a VideoToolKit for image-based preprocessing, which analyzes and extracts visual features and semantic descriptors, then builds the video database scheme used to represent and index the video. The video preprocessing partitions videos into meaningful segments using a shot cut detection algorithm with automatic threshold determination, and these shots are the units for query and retrieval of video source. Representative key frames are extracted from each shot, and visual features that identify shot content are extracted from each frame. Processing includes camera motion classification, low-level visual feature extraction, object detection, contour-based temporal tracking, and spatial and temporal segmentation. In addition to storing frame-based indexing information, aggregate "per shot" average, minimum, standard deviation and variance values (across all frames) are computed and stored. MPEG7  has become the worldwide standard for multimedia content descriptors, and it is an integral part of VDBMS feature representation. Our preprocessing extracts nearly all low-level features defined by MPEG7, including color histogram (HSV, YUV), dominant and scalable color, color moment and layout, texture tamura and edges, homogeneous texture, motion and edge histograms.

The VDBMS system has been built on top of an open source system consisting of Shore, the object storage manager developed at the University of Wisconsin, and Predator, the object relational database manager from Cornell University. The VDBMS research group developed the extensions and adaptations needed to support full database functionality for "video" as a fundamental database object, including high-dimensional indexing, multimedia query processing to support image similarity search, buffer management of large objects with real-time constraints, and online video stream management with real-time constraints. Figure 3 shows the original components from the Shore/Predator system and identifies both current and proposed VDBMS adaptations of the original system. Most existing system components were significantly affected by introducing and supporting video as a fully functional type.

To support feature-based image queries, VDBMS performs real-time searching against the high-dimensional feature vectors resulting from feature extraction preprocessing. Image information extracted from each frame of a video occupies a vector of cumulative dimension approaching 1000; the magnitude of this indexing information required an entirely new multi-dimensional indexing structure for image similarity searches. Our concept is based on Berkeley's Generalized Search Tree (GiST) implementation of the SR-tree as the indexing technique, where the nearest neighbor (NN) search operator uses an index access path created by running an incremental NN search query on the SR-tree.  The VDBMS Index Manager extends the indexing capability of Shore by incorporating the GiST SR-tree as the high-dimensional index. A "vector" ADT was created for the high-dimensional visual feature fields, and an instance of the SR-tree is used as the access path in feature matching queries. We have developed a real-time Stream Manager that is responsible for the streaming of video data directly from the database with real-time constraints at a predetermined rate. The streaming component admits, schedules, monitors and serves concurrent video streams requests periodically, issuing requests to the Buffer Manager to guide underlying buffer caching policies for stream requests. The Stream Manager has well-defined interfaces with the query engine, buffer components, and the E-ADT interface. Extended buffer management handles multiple page requests with segment allocation instead of the traditional page-based approach for more efficient handling of the huge volumes of stream data. Our methods for handling extended storage hierarchies support real-time access to buffer and disk storage. Different caching levels have been implemented for frequently accessed data, reducing the number of references to disk storage and minimizing latency associated with the start of streaming. The Query Manager was modified extensively to handle the new high-dimensional index schemes and to support multimedia feature-based queries and video query operators. New query types were introduced to support the video-related operations (query by image, visual feature nearest neighbor search) involved in generating, optimizing and executing query plans. A query interface supports end user query, retrieval and presentation (Figure 4.) Users can pose frame- or shot-based image queries, query by motion information, query by keywords, or SQL statements. Shot-based queries use aggregate feature similarity matching. Results are presented as an image filmstrip, using key frames to represent retrieved shots. Users select a key frame to stream the video shot directly from the database to the client media player.



Select vs.VideoId, vs.ShotId, vs.IconFrame, vs.StartFrame, vs.EndFrame from (select [illegible]

Find 5 Results

Step 1

1/5
VideoId : 1
ShotId : 10
StartFrame : 2987
EndFrame : 3253

2/5
VideoId : 1
ShotId : 47
StartFrame : 13110
EndFrame : 13173

3/5
VideoId : 1
ShotId : 25
StartFrame : 8821
EndFrame : 8831

4/5
VideoId : 1
ShotId : 41
StartFrame : 12709
EndFrame : 12755

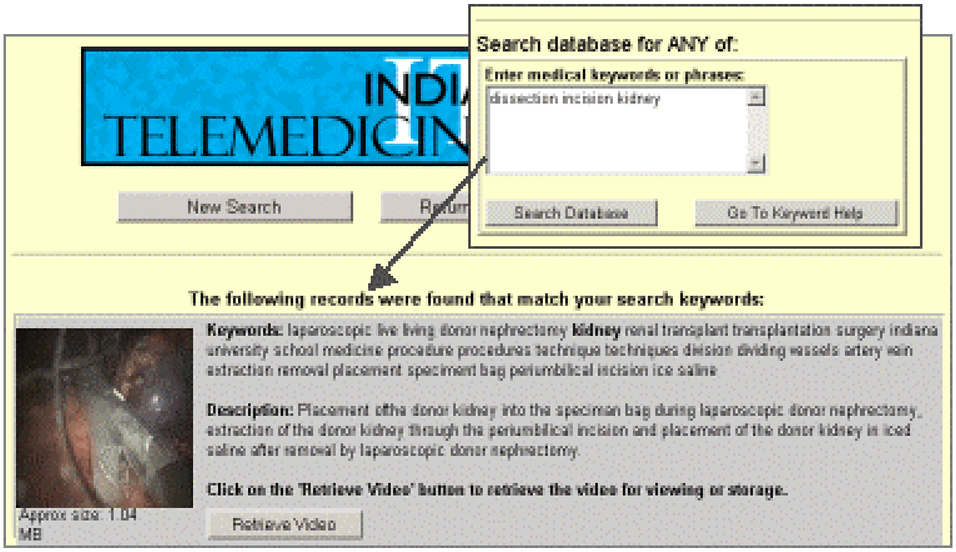
**Figure 4. Research query interface for VDBMS. Clients connect to the database server which resides on a SUN Enterprise 450 with 4 UltraSparc II processors and a Sun RAID array located at Purdue University.**

Through the MSI infrastructure grant, we have achieved significant results related to multimedia database research. We have designed an architecture for managing multimedia documents with Quality of Service (QoS) guarantees for real-time distributed multimedia systems. We are exploring different approaches that allow mapping of the user-specified Quality of Presentation (QoP) parameters to QoS requirements for different system components of the overall architecture, including network, storage, security, and server subsystems. For network QoS, we are investigating the theory and implementation of aggregate flow scheduling and integrated communication-computation scheduling. We have designed novel data placement and scheduling schemes for the physical storage management of multimedia documents. We have designed an adaptive security architecture (with components for monitoring, control and actuation) and have developed a distributed Denial of Service (DoS) attack prevention analyzer. VDBMS research is connected with investigations into the multimedia database server subsystem component, demonstrating the continuity and interrelated nature of our research approach. The design and development of the MSI infrastructure is ongoing.

Research and development efforts for VDBMS have produced some of the most advanced techniques and models currently available in video database management. The VDBMS framework was the foundation of the development effort for EduMed: WebVideo for Medical Education, a prototype system created through the ITI 21st Century grant.

**Digital Medical Archive and the EduMed Query Interface**.

ITI was established to leverage the latest research and technology in multimedia databases for health care industry applications. This technology was applied to trial environments for medical education and remote diagnostic medicine supporting 1) management and use of multimedia data, 2) annotations of medical video data with text and audio commentaries by physicians, 3) content-based search and retrieval using indices built from the annotations, and 4) coordinated storage and retrieval of video data from remote sites. The goal for a trial environment in medical education was the creation of a small-scale indexed medical archive and the development of the EduMed prototype query interface to support archive search and discovery based on semantic keyword queries (Figure 5.) Functional specifications were developed in collaboration with faculty from the Indiana University School of Medicine. The current system provides web-accessible storage, access, retrieval, and streaming for an indexed MPEG-encoded medical video archive based on semantic video content representation. Faculty can prepare and deliver video clips through keyword query, retrieval and streaming video presentation.



Search database for ANY of:

Enter medical keywords or phrases: dissection incision kidney

The following records were found that match your search keywords:

**Keywords:** laparoscopic live living donor nephrectomy **kidney** renal transplant transplantation surgery indiana university school medicine procedure procedures technique techniques division dividing vessels artery vein extraction removal placement specimen bag periumbilical incision ice saline

**Description:** Placement ofthe donor kidney into the specimen bag during laparoscopic donor nephrectomy, extraction of the donor kidney through the periumbilical incision and placement of the donor kidney in iced saline after removal by laparoscopic donor nephrectomy.

**Click on the 'Retrieve Video' button to retrieve the video for viewing or storage.**

Approx size: 1.04 MB

**Figure 5. EduMed query interface. Faculty build video clip collections with query results obtained through search and discovery of the digital medical archive**.

Other useful results from the ITI project will be incorporated into the system developed through the research proposed herein. For example, ITI diagnostic telemedicine activities at Clarian Health medical center will contribute new and significant media forms, such as digitized heart and lung sounds recorded during patient teleconsultation. The current capabilities of EduMed will be significantly expanded as a result of advances in the proposed research areas.

## A Distributed Multimedia Digital Libraries for Medical Education

The task of creating an extensive archive of multimedia medical data involves the acquisition of high-quality, relevant medical videos and their digital compression, pre-processing and storage. Medical video sources have already been identified[3] and our MPEG1 compression uses advanced off-the-shelf commercial technology that is capable of creating MPEG in real-time with excellent control over the rendering process[4]. The VDBMS VideoToolKit handles video content preprocessing and database insertion. Our research and development will focus on advances that support the accessibility and usability of the multimedia data contained in the medical archive.

[3] Medical Education Resource Program (MERP), Indiana University School of Medicine is our principal source for video and image data. Several faculty members intend to produce their own videos for the medical archive.

[4] Division of Instructional Services, Purdue University provides our compression services through MediaPressPro. Real-Time MPEG Compression. http://www.media100.com/product/mediapress

This research encompasses two principal areas: advancing the underlying multimedia database technology and investigating issues related to the development of web-based agents for intelligent processing of high-level “concept” queries and secure access of distributed multimedia data.

## Advancing Multimedia Database Technology

The proposed extensions to multimedia database technology address the processing of complex multi-feature queries for image similarity matching, advanced stream handling, and extended storage hierarchies. Each of these areas presents significant research challenges.

**Optimal Aggregate Ranking for Multi-feature Queries.** In multi-feature image similarity queries, users present a sample image and query the database for images “most similar” to the sample based on some collection of visual features. Although database images can easily be ranked for each feature separately, results for multi-feature queries must be presented to the user in a combined similarity order. The query evaluation model used for a similarity search does not generally return a collection of exact matches, but rather a ranked collection of results with a score attached to each result. The aggregate score for a given result is obtained by combining the scores of several atomic similarity rankings, where an atomic ranking is based on a single feature or attribute of the database object. Single-feature similarity queries in multimedia retrieval are quite standard The challenge for multi-feature similarity search is the efficient, effective ranking of results based on an overall aggregate score, using several atomic rankings as input. Many algorithms have been proposed in the literature to address aggregation ranking, notably Fagin's algorithm, the TA and CA algorithms, the Quick-Combine algorithm, the multi-step aggregation algorithm, and the No Random Access (NRA) algorithm. Two alternatives exist for implementing rank-join algorithms for databases: table functions or encapsulation in a physical query operator. Since there is no straightforward method for pushing query predicates into table functions their performance is severely limited. Implementing a rank-join algorithm as a pipelined query operator, however, is very appealing for query optimization. An operator allows for efficient handling of nested joins and views, with greater flexibility in generating candidate execution plans.

For a rank-join algorithm to be implemented as a pipelined query operator, the algorithm must have two key properties. First, the algorithm must be incremental, that is, it must not depend upon knowing the number of required results beforehand, but rather must provide the next result when called. Second, the operator must support pipelining. For a query operator to be part of a pipeline, the output of one operator should be valid input to the next operator in the pipeline. Rank-join algorithms that depend on the availability of random access to the inputs cannot be realized as pipelined operators, since random access is not possible for input that arrives as output from another operator. Though the NRA algorithm assumes only sorted access for the input streams, it still cannot be pipelined. This is because its output objects do not have exact scores associated with them, and NRA cannot accept this as valid input to another NRA execution. We intend to modify the NRA algorithm so that it can be realized as a pipelined query operator. We will then encapsulate and implement the modified algorithm as a logical query operator in VDBMS. The new operator will support image similarity matching with respect to multiple features, returning the best globally ranked results. In VDBMS, each database frame image is represented as a point in high-dimensional space, and the image similarity problem will be transformed to a nearest-neighbor query on our high-dimensional feature indexing structure.

The key contribution of the new rank-join operator is the modification that allows the joining of ranked inputs to be realized as a binary pipelined query operator, which can then be integrated into query pipelines to support a hierarchy of join operations. The operator transforms the optimal theoretical solution into one that can be implemented in a practical database engine. The efficiency of this operator will be tested within the VDBMS framework.

**Advanced Stream Handling.** The nature of video data, whether processed by the database to answer queries or delivered to the client from the database, requires the extension of underlying database technology to support stream processing for large objects. In this proposal, we address stream processing in three critical areas: the investigation of query operators capable of joining multiple infinite data streams together with stored data to answer user queries, the use of database access control methods that alter the video data as it streams to the client in order to protect privacy or ownership, and the reduction of latency and disk I/O in streaming video through improved buffer management policies that consider the inherent connection between query results and client stream requests. These issues will be addressed by establishing the definition and model of the data stream on a suitable level of database abstraction and then identifying the representation of stream characteristics, the distinct types of stream operators, the necessary operator algebra, and the associated operator semantics.

We define a stream as an infinite sequence of data items, where items are appended to the sequence over time and the sequence is ordered by time stamp. Accordingly, we model each stream data item as a binary tuple <*v*,*t*>, where *v* is a value (or set of values) of the data item, and *t* is the time at which this item joined the stream. The data item can be a single value or a vector of values, and each value can be a simple or composite data type. Time *t* is our ordering mechanism, and the time stamp is the sequence number attached to each new data item. A sensor is any data source that is capable of providing infinite streams of data, either continuously or asynchronously. Streams of data are created by many monitoring and recording devices; in our system we are interested in streams produced by medical devices that transmit sounds (e.g., heart and lung) from the human body during teleconsultation and telediagnosis. These sounds can be recorded, digitized, archived and catalogued. They provide significant value for teaching and, more importantly, our proposed stream query processing can be applied to both streaming and stored tele-health data to support efforts for advancing automated diagnosis.

Our proposed stream processing framework will operate as follows. Sensors will provide raw data to the pre-processing unit, which prepares the stream for database operations by executing feature and content analysis/extraction on the incoming stream (using the VideoToolKit.) The stream processing system will keep information about the streaming sensors in a local database; these may include error probability associated with incoming data, statistical distribution of stream values, headers associated with groups of streaming items, and the rate of streaming. The Stream Manager will act as a buffer between stream query processing and the source streams, and will be responsible for serving requests by the query engine to fetch new data. Data will be collected from the input streams into the stream buffer. Interaction between the Stream Manager and the input streams can be either push or pull based. In the push-based approach, the Stream Manager initiates the stream request from the input sensor and monitors the arrival of the processed input stream. In the pull-based approach, the Stream Manager actively requests data from the input source, but only if there is a pending request for this data from the Query Manager. The post-processing unit exports the output stream to the requesting client. A stream-enabled query manager will be responsible for executing the new query operators as well as optimizing the execution plans that involve groups of query operators. Our investigations will include the development of query processing algorithms for stream operators, design of the execution framework for queries with multiple operators, optimization of resource usage during execution (both storage and processing), and support for multiple online and long-running queries.

**Query operators for joining multiple infinite streams and combined sensor/stored data**. Stream size is unbounded since data continues to be added to streams over time, and query processing systems that operate on stream data must take this into account. For the most useful queries, multiple streams of data must be joined in order to process the query. This problem has recently generated a great deal of interest in the research community. Several join techniques have been proposed. The approaches described in Ripple Join, double pipelined hash join, and XJoin require processing the entire contents of the stream encountered thus far (the stream *prefix*), as well as all incoming tuples. Since these join methods do not consider restricting the prefix, they cannot handle infinite streams. Techniques which update the prefix of a stream to maintain a window over recent tuples are more appropriate for stream processing. While the band join technique addresses the problem of joining two relations (of fixed sizes) for values within a "band" of each other, this work does not address more than two relations, and it assumes access to the entire relation. In COUGAR, the system can execute queries over both sensor and stored data, but the notion of the stream as an infinite data source is not addressed. The FJORD project proposes a framework for queries over sensor data as well as stored data using a symmetric hash join, but their technique only applies to special cases of the join (equi-join and no joining between two streams). The window approach has been used to execute aggregate operators over data streams and for self-joining a single stream however these techniques cannot be extended to join multiple streams.

We propose a class of window-join algorithms that will join multiple infinite data streams and apply to any join condition. Our window-join defines a sliding window defined over time intervals, and only tuples within this window are considered for the join. We first propose an algorithm for a fixed-size window join. We must avoid repeated iterations over non-window-related tuples, and we will do this by verifying window constraints between the input streams and then updating join buffers to contain only eligible tuples. We will also address the removal of old tuples that have moved outside the window. We plan to implement this algorithm in the pipeline query plan, using both nested-loop and hash techniques. The fixed-size window join will then be adapted to evaluate variations of the window-join which address unequal windows sizes and missing window constraints. All algorithms will be implemented in the VDBMS stream processing system where the effectiveness and efficiency of the algorithm can be tested.

**Selective access mechanisms for confidentiality and privacy in streaming media.** Privacy is a major concern in the dissemination of any data, and this is especially so in the case of shared medical information that includes images and videos of patient consultations, examinations, and surgeries. Clearly, patient privacy is of paramount concern, but patient identity and other patient-related information must be protected in a way that does not render the video content useless. Our proposed method addresses this problem by establishing an access control model that provides selective access to data according to an established security policy. Our model and its associated mechanisms take advantage of ongoing work in access control, video semantics, and access patterns. The first layer of access control will allow different classes of users to receive different authorizations for the same set of data, based on user qualifications or characteristics rather than on identity. A second layer will support content-dependent authorizations for video objects. Access is given or denied to a class of users based on the semantic contents of the video. The third layer will be based on the hierarchical structure of videos (stream, scenes, frames, objects) and requires a mechanism that supports varying granularity levels for authorized objects.

Several results from the literature extend conventional database access control models to new data types and associated functions for authorization management. Such efforts include temporal authorization models and extended authorization models for relational databases. These models, however, are not adequate for the protection of information in a video database system, mainly because conventional authorizations are specified in terms of user and object identifiers, rather than in terms of user profiles and object content. A proposal by Kumar and Babu can hide entire frames for specific classes of users, but has no support for sub-frame or object restriction. Furthermore, even though their approach partitions users into categories, it does not support authorizations containing predicates against user profiles. Content-dependent access control has been addressed both in relational and object-oriented DBMS through the use of views. Content-based access control is enforced by specifying conditions against attribute values of data objects. In contrast, the nature of video objects requires content-dependent access control to be based on the semantics of the objects, rather than on attributes characterizing them.

Our proposed access control model is based on video semantic content and the structured composition of video data, and presents a natural flexible way to express authorization for video data. The unit of authorization is a video element: either a sequence of video frames or a video object that appears as part of a frame, such as the face of a patient conferring with a doctor in an examination room. Components of the model are video elements and potential users. Video elements are specified either explicitly by their identifiers or implicitly by their semantic content, while users are characterized by their credentials. The proposed algorithm determines the authorized portions of a video that a given user may receive, given the user credentials and the video content description. Specific features of the proposed model include access control specification for video data objects based on their content (rather than identifiers), flexible specification of authorization based on the notion of user credentials, and varying granularity of authorization objects ranging from an entire video to video segments to specific portions of video frames. In order to support content-based access control, we will provide a general framework for semantic content analysis that supports either the semantic or the visual feature descriptors of video content. Integrating the control model into VDBMS will be achieved by specifying a set of authorization rules and control procedures. Authorization rules describe *who* is allowed to access *what* in the database, and control procedures will deploy the rules on a transactional level.

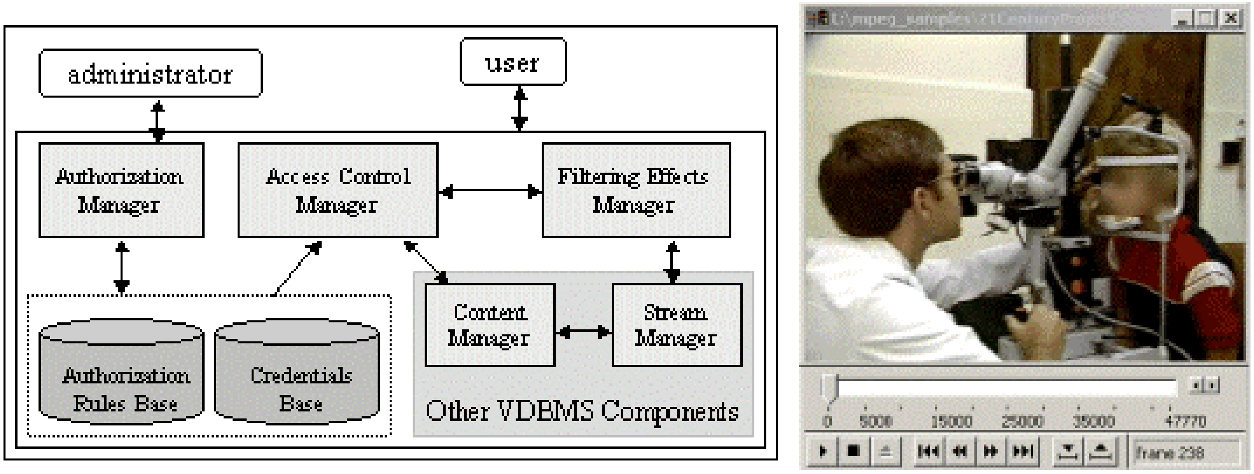


**Figure 6. Framework for the VDBMS access control system (left) and sample of a frame from the streaming video, where the patient's face is blurred to protect identity (right.)**

The subject in an authorization model represents the entity attempting to access the database. We incorporate information about user characteristics and profiles by adopting the notion of credentials, which identify a set of user-specific security-relevant information. Credentials extend the expressive power of security specifications and allow the model to better relate subject qualification and characteristics with the semantic contents of video objects. The credential mechanism provides a simple language for formulating credential expressions against which user characteristics are matched. Evaluation results in the set of users that satisfy the expression requirements. Whereas subject specification is independent of the underlying data model, object specification is tightly coupled with it. We intend to use a logical representation for supporting content-based access control. The specification of an authorized video object in our model is based on the logical video elements: logical video, logical video segment, and logical object. These video elements can be specified either directly, by providing their identifiers, or through a set of concepts. In simplest form, a concept might be the keywords from the video annotations. Concepts specify semantic information about the actual contents of a set of video data objects, and they determine how to restrict the access to videos dealing with a specified content. Figure 6 shows the proposed access control framework along with a sample of an altered frame received by an end-user during streaming. In the sample, the user's view of frame content is restricted within the operational context for subject and object authorization.

**Search-based buffer management policies for streaming continuous media**. Continuous-media servers that support content-based query and retrieval use a main memory buffer to store the requested media streams before sending them to the user. Caching parts of media streams that may be referenced in the near future enhances streaming performance by reducing the number of references to disk storage and minimizing delay associated with the start of streaming. Precise caching decisions are difficult to make, however, since they depend on future knowledge about "expected" streams. We propose a new buffer pre-fetching and replacement policy that is based on integrating knowledge from the query and streaming components. The policy will use both current and expected video streams to predict streaming requests, where expected streams are those corresponding to the top-ranked query results from the Query Manager. This concept is based on the inherent connection between query results and streaming, i.e., users generally choose to stream their highest-ranking query results. Many factors must be considered when basing pre-fetching or replacement decisions on search results. Most importantly, streaming decisions are probabilistic, and this uncertainty must be represented in our scheme. In addition, since pages for current as well as expected streams must now share cache space, the processing associated with space balancing will increase overhead.

Buffering policies for media streaming have been investigated in several studies. Using fixed time-displacement between pre-fetching requests has produced a memory efficient pre-fetching scheduler. Several replacement policies have been investigated for target applications designed solely for media streaming, such as video on demand (VOD) applications. Ozden, et al. introduces two replacement policies based on caching data expected for play by other streams within the shortest time period. These algorithms produce effective buffer hit ratios for long streams (lasting more than half an hour) and for streams frequently requested by clients. The Use &Toss strategy has been suggested for sequential access patterns, but this strategy does not consider caching the page after using it. In each of these studies, the focus is on systems whose only function is media streaming. Furthermore, most of the previous studies consider VOD-like applications, where the video data set is small and streaming lasts for a long time. Our application domain is quite different - the data set is large and the streaming time is short. The key concept in our proposed policy is a table identifying expected streams, which is built by the Stream Manager using information about query results delivered from the Query Manager. The table will be maintained by the Stream Manager and used to identify data blocks that should be pre-fetched or replaced. The application of the relationship between searching and streaming to improve streaming efficiency has not previously been addressed in the literature.

The VDBMS Stream Manager stores information about concurrent streams in its own data structures. When a new stream request is admitted, the portion of the stream to be accessed is declared. The information is recorded as a tuple of the form *T = (CM#, start, end)* where *CM#* is a unique identifier for the stored stream, and *start, end* indicate the starting and ending block numbers of the streamed data. For current streams, the manager keeps the tuples in the lookup table $T_{cs}$ and modifies the starting block for serviced stream at the end of each period to track the currently streaming blocks. In our proposed policy, when the Query Manager replies to a user query with a set of candidate results, the Stream Manager will receive a copy of the search results and consider the top-ranked results as likely for future streaming. These expected streams will be stored in the lookup table $T_{es}$. The information in the two tables will be used with different confidence levels, since $T_{cs}$ describes actual system status, whereas $T_{es}$ represents only an expectation of future references. Our core buffer manager uses the Generalized CLOCK (GCLOCK) replacement algorithm for buffer page replacement, and associates a *keep-weight* counter with each page. GCLOCK views the buffer pool as a circular list with a pointer identifying the next page to check for replacement. When a page is to be allocated, the pointer scans the unfixed pages searching for a page to allocate, and continues to decrement the counters until reaching a page with counter equal to zero. In the current policy, as page $P$ is streamed by request $R_i$, the Stream Manager determines whether any concurrent streams reference $P$ by checking $T_{cs}$ for a request $R_j$ with the same CM# such that $P$ is within the start-end block interval given by $R_j$. One match is enough to keep the page in the buffer pool, and if a match is found, the Stream Manager sets *keep-weight>1* to inform the buffer manager that the page should not be replaced for some time. In our proposed replacement policy, if no matching entry is found in $T_{cs}$, then $T_{es}$ will be checked. Caching will be restricted to the first segment of the expected stream to avoid caching pages that remain unreferenced for long periods. The keep-weight for a match in $T_{es}$ will be less than that established for $T_{cs}$ to reflect that we are less confident about expected-streams than actual streams. If no match is found in either table, the Stream Manager will set *keep-weight =0*, indicating that the page can be replaced immediately.

In our proposed pre-fetching policy, the Stream Manager will utilize any fraction of the period unused by current streams to pre-fetch the first segment of top-ranked expected-streams into memory. While serving the current stream requests, the Stream Manager will continue to track the utilization of the streaming period. If it detects that the number of serviced streams is less than the maximum possible, it will consult $T_{es}$ and choose one of the expected-streams for pre-fetching into the memory buffer. Afterwards, it will immediately unfix its pages and set *keep-weight>1*. In this way, the segment will be kept in the buffer pool for some time before being replaced. The stream manager will loop around the expected-streams, bringing in the first segments of top-ranked streams before serving lower ranked ones. If a pre-fetched segment turns out to be unreferenced, it will be aged in the buffer pool until the *keep-weight* is reduced to zero, forcing the segment to be replaced out of the pool. If an expected-stream becomes an actual request, most of the pages in the first segment will already be cached in the buffer pool, so that the number of references to lower level storage will be significantly reduced. Extensive experiments in a heavy workload environment will be undertaken to evaluate the performance of the proposed replacement and pre-fetching policy.

**Extended Storage Hierarchies.** We will extend the underlying technology of database storage management to support multiple caching levels on buffer and disk storage to optimize system performance in accessing frequently referenced data. We plan to introduce a new tertiary volume object and create a Tertiary Storage Manager to manage real-time operations on tertiary resident data, making it directly accessible to the VDBMS system.

Read/write requests will continue to be processed by accessing data pages from the buffer pool, and the Buffer Manager will still service requests by retrieving pages from disk when not found in the buffer pool. In the proposed model, the Buffer Manager will not be aware (at this level) of whether data is on disk or in tertiary storage. Service requests will continue to be sent to the Volume Manager, which controls the activity of the physical disk, but it will be modified to process the new tertiary object and support new tertiary-volume-specific computations. Two new modules, the Tertiary Storage Manager and the Cache Disk Manager will be implemented underneath the Volume Manager. A dedicated disk partition will be used to cache hot items stored in tertiary storage, and housekeeping of the cache items will be handled by the Cache Disk Manager using a hash table implementation. Pages not found in cache will be retrieved from tertiary storage by the Tertiary Storage Manager, which will communicate directly with a remote PowerFile Manager controlling the commercial tertiary storage devices (e.g., daisy chained DVD jukebox servers.) Requested data will sent through the local network with TCP using block (128K) instead of page (8K) units due to the excessive overhead associated with servicing requests. Blocks will be copied to cache disk and the first page will be sent to the buffer pool. Cache disk will use a least recently used algorithm for replacement, and the search-based policies described previously will be incorporated for pre-fetching from tertiary storage. The existence and operation of the Tertiary Storage Manager will be transparent to the object relational database interface layer. We will measure and assess the performance of this storage model, and experiment with scheduling and data placement algorithms to enhance real-time performance.

**Additional Areas of Research.** To further advance the preprocessing of multimedia data, we plan to investigate new low-level image and high-level semantic processing techniques. For this research, we will leverage collaborations with senior personnel who have done previous work on VDBMS image processing algorithms, in particular for feature analysis and extraction. We also plan to establish MPEG7 document compliance through the import and export of MPEG7 documents generated using Multimedia Description Schemes (MMDS) for high-level and low-level feature information. We propose to develop an MPEG7 wrapper to import MPEG7 documents, parse document features and map the MMDS descriptors to our database feature schema. The wrapper will also export features information from VDBMS as MPEG7 documents.

## Expanded Capabilities for the EduMed Prototype System

A system that delivers intelligent, unified access to distributed multimedia data requires the development of a number of related web-based components. Our components will be implemented using an agent platform that supports distributed agent processes, providing the system with a base for communication services, mobile computing power and dynamic information retrieval. We plan to investigate the Grasshopper agent platform from IKV++ Technologies AG for implementing our distributed agent-based components. Grasshopper is built on top of a distributed processing environment that integrates the traditional client-server paradigm and mobile agent technology. The environment facilitates the creation of agents, transparently locating them and controlling their execution. Grasshopper agent management facilities are responsible for communication services for all remote interactions, registration services to track currently hosted agents, managements services to monitor and control agents, transport services to migrate agents across the network, basic security services to protect remote interactions, and persistence services for possible recovery operations. We will focus on building agent-based components that support user concept query processing, secure data access, and processing of retrieved multimedia results for presentation.

Query processing that can effectively address multiple data types (video, image, text, audio) by formulating appropriate requests to one or more target databases belonging to a collection of heterogeneous systems is still an open research question. In addition, most end-users are not familiar with the contents of the digital archive they are searching against, and they may not know exactly what they are looking for. The notion of 'concept query' offers a partial solution to these problems. Concept query processing will apply intelligent interpretation and translation to user queries that are posed at a high-level. The high-level queries will be translated into multiple content-based queries, each addressed to one or more relevant information domains. Processing is supported by a knowledge base of content-summary information that is extracted from member databases, then filtered, cataloged and stored. Our knowledge base will initially be created using summary feature and content indexing information extracted both from VDBMS and from a small number of information domains outside the VDBMS environment, including the Internet. The knowledge query agent will access the knowledge base to interpret, formulate, and direct appropriate queries to the appropriate information domains.

Allowing users to access digital video archives containing sensitive medical data through a web-based interface introduces serious privacy and security concerns. Issues related to secure communication between client/server, authentication of requests, and content protection must be investigated. Data access control at the data source was addressed earlier. We now address issues related to the distributed implementation of the security mechanisms. These issues can generally be divided into three categories: 1) communication security, which ensures that the data is not accessible to or corrupted by anyone other than the sender and recipient, 2) authentication, which certifies that the sender and receiver are who they claim to be, and 3) access control, which verifies that the parties have the right to view the information. Existing standards (e.g., SSL) support communication security. There are mechanisms for authentication, such as public key cryptosystems, although managing authentication for the number of users who will access this system over the web does pose new challenges. Access control in a distributed environment, however, can be unmanageable. While technology for access control exists (e.g., SQL grant/view mechanism), managing and maintaining fine-grained access control in a distributed, web-based environment is an administrative nightmare. We will investigate the following research issues for the proposed security component:

- Permission semantics suitable for distributed systems: Information permissions should be granted once, not once for each copy or variant of the same information. Physicians or faculty authorized to view sensitive patient examination videos shouldn't need to request separate authorization to access similarly protected data stored in separate information domains.
- Managing permission sets on each object: The content-based access restriction described previously must be extended for application in a distributed environment.
- Integrating information releasability with access controls: Some policies are difficult to enforce with access controls: Can I show data I receive to others? Can I put it on my home computer?
- Protection requirements: A policy specification should include a technology-independent description of how strong enforcement must be, i.e., of tolerated level of vulnerability. Queries that retrieve general information describing group of patients may be okay to send "in the clear", individual patient information isn't.

Several additional areas of development are needed to ensure the usability of our system. Both semantic video preprocessing (annotations) and concept query processing must have a well-defined interface to an established medical concept ontology, such as MESH or the UMLS Metathesaurus from the National Library of Medicine. This is a challenging task, but it is essential to standardize the medical terminology used throughout the system. Access to information domains outside of the VDBMS system will require vocabulary translation services as part of concept query processing. Finally, we will take a limited approach to the integration and presentation of retrieved multimedia data to the user. Our current VDBMS system supports user navigation through video and image results. The presentation of combined video, image, text and audio data will require new visualization and navigation methods that convey the content, context, source, and relationships for combined data types. We propose to develop a multimedia browser that can display multimedia results and identify their interconnections.

## Archive Deployment Environment

We plan to deploy the medical education query and retrieval system at the eight centers and central campus of the Indiana University School of Medicine. A dedicated SUN Enterprise 450 server located at the Lafayette Center for Medical Education will support the digital multimedia medical archive.