

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE Nov 95	3. REPORT TYPE AND DATES COVERED Final 1 Aug 95 - 31 Jul 96	
4. TITLE AND SUBTITLE Workshop on Multimedia Database Systems			5. FUNDING NUMBERS DAAH04-95-1-0423	
6. AUTHOR(S) V.S. Subrahmanian Satish K. Tripathi				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) University of Maryland College Park, MD 20742			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) U.S. Army Research Office P.O. Box 12211 Research Triangle Park, NC 27709-2211			10. SPONSORING/MONITORING AGENCY REPORT NUMBER ARO 34409.1-MA-CF	
11. SUPPLEMENTARY NOTES The view, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution unlimited			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) DTIC QUALITY INSPECTED 4 The First International Workshop on Multimedia Information Systems was held in the Doubletree Hotel National Airport, Arlington, Virginia from September 28th to 30th, 1995. The workshop was chaired by Prof V.S. Subrahmanian and Prof. S.K. Tripathi. The workshop was sponsored by the Army Research Office and the University of Maryland Institute for Advanced Computer Studies. There were about 60 participants from academic institutions, research laboratories, the Army and the industry.				
14. SUBJECT TERMS			15. NUMBER OF PAGES	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL	

19960212 102

Final Project Report
November 21, 1995

Workshop on Multimedia Database Systems

ARO Grant No. DAAH-04-95-10423

Principal Investigator: V.S.Subrahmanian
Co-PI: Satish K. Tripathi
Department of Computer Science
University of Maryland
College Park, MD 20742.

1 Overview of Workshop Proceedings

The First International Workshop on Multimedia Information Systems was held in the Doubletree Hotel National Airport, Arlington, Virginia from September 28th to 30th, 1995. The workshop was chaired by Prof V.S. Subrahmanian and Prof. S.K. Tripathi. The workshop was sponsored by the Army Research Office and the University of Maryland Institute for Advanced Computer Studies. There were about 60 participants from academic institutions, research laboratories, the Army and the industry. As many as 10 persons had to be denied registration in view of the space constraints in the convention hall. The workshop was inaugurated with a welcome note by Prof. S.K. Tripathi.

The technical sessions started with an invited talk by Prof P. Venkat Rangan, University of California, San Diego, on *The MPEG Multimedia Stream Architecture*. The MPEG standard and the specification of continuity and synchronization within the MPEG standard, and a proposal for their implementation in a distributed multimedia environment were covered by the talk.

The next session was chaired by Prof. Rich Gerber. The first talk was by Prof R. Muntz, University of California, Los Angeles, on *Virtual World Data Server*, designed for efficiently storing and retrieving large amounts of spatially distributed heterogeneous data. The server has been designed to provide service for multiple concurrent real-time 3D interactive sessions. In the following talk on *Multimedia Applications in an IP over ATM Environment*, Dr D. Kandlur, IBM TJ Watson Research Center, proposed an approach for ATM deployment in the TCP/IP environment that minimizes new development and impose no new requirements on ATM. Prof M. Herzog, University of Vienna, speaking on *Multimedia Information Systems in Open World Domains*, argued for the use of multimedia components in within information systems to provide guidance in handling exceptions that happen in the *open world*. The last talk of the session was by Dr P. Mishra of AT&T Bell Laboratories on *Network Protocols For Wireless Multimedia Access*. He described an architecture for handling communication problems unique to wireless network and for allowing internetworking of two types of networks.

The session was followed by an invited talk by Prof Jim Kurose, University of Massachusetts, Amherst, on *Multimedia Networking*. In this talk, the range of requirements for multimedia applications such as teleconferencing and video on demand, were identified. Then the open challenges and research issues including receiver-initiated control and adaptivity and the need to provide Quality of Service guarantees were discussed.

The second paper session was chaired by Prof. Jajodia, George Mason University. It began with a presentation by Prof P. Buneman, University of Pennsylvania, on *New Languages for the Integration of Heterogenous Data Sources*. The next talk was by Prof. V.S. Subrahmanian, University of Maryland, College Park, on *Foundations of Multimedia Information Systems*. A theoretical basis for multimedia information systems using the mathematical model of a *media instance* was presented as part of the talk. Dr Hemant Kanakia, AT&T Bell Laboratories, speaking on *Road Runner : An Operating System for Multimedia Applications*, proposed a focus on the inter-process communication instead of the processor-centered common view, in building new OSs.

The third paper session started on the morning of September 29th, 1995, chaired by Dr P. P Mishra. Prof V. Gudivada, Ohio University, proposed a query reformulation algorithm based on the functional dependency between each image attribute and the user's relevant feedback using atheoretical framework referred to as rough set theory.

This session was followed by an invited talk by Prof J.D. Ullman, Stanford University, on *Research into Multimedia Database Systems*. He identified several areas in the database technology that will be important : Management of tertiary storage, development of type systems for special kinds of information, query systems for multimedia information, and support for varying qualities of service.

The fourth paper session was chaired by Dr. Sherry Marcus. Prof E. Bertino, University of Italy, speaking first on *Query Refinement in Constraint Multimedia Databases* discussed approximation strategies in the framework of constraint databases. Mr. K.S. Candan presenting the paper on *Advanced Video Information Systems : Data Structures and Query Processing*, described how video data may be organised using modified spatial data structures as to facilitate queries.

Prof. Anil Nerode, Cornell University, presented his work on *Hybrid System Methods for Distributed Multimedia Systems*. This was followed by the talk by Prof. Richard Gerber, University of Maryland, on *Benchmarking Digital Video*. He presented the results and analysis of the experiments on multimedia applications conducted on two Apple Macintosh platforms : Quadra 950 and Power PC 7100/80. Speaking next on *Multimedia Information Systems*, Dr. Sherry Marcus, 21st Century Technologies, described a mathematical structure on the set of features/states of media objects. Using this notion of a structure, she presented the definition of indexing structures for processing queries and methods to relax queries when answers do not exist.

The fifth paper session was chaired by Professor E. Bertino, University of Milano. The first paper was presented by Dr. Lokendra Shastri on *Integrating a Scalable Rapid Reasoning System with a Database System*. He proposed methodologies for

integrating a limited class of first-order inference on a scalable and efficient knowledge representation and inference system, with a relational database system. The next paper was by Professor N. Rishee, Florida International University *On Storage and Retrieval of Generalized Spatial Data*. He presented the development work on a prototype high-performance semantic spatial database management system. The last talk was on *Content-Based Retrieval of Pictures and Video* by Dr. A. P. Sistla.

The last day of the workshop, September 30th 1995, began with an invited talk by Dr. Robert Allen of Bellcore on *Multimedia Information Systems*. He discussed the key features of electronic text-document browsers and the ways these might be applied to multimedia documents. Some of the features included structured views, linking and indexing.

The last paper session was chaired by Professor V. Gudivada, Ohio State University. Professor S. Ghandeharizadeh of the University of Southern California spoke first on *A Framework for Conceptualizing Structured Video*. He proposed a conceptual model of structured video consisting of an object space and a name space. The object space represents the rendering aspects of video and the name space represents the user interpretation of the video. The next paper was on *Object Oriented Modeling and Querying of Multimedia Data* by Professor A. Ghafoor of Purdue University. He utilized the n -ary relations among multimedia objects to propose a query language that provides facilities for declaration of multimedia data, document composition and imprecise retrieval, spatial/temporal composition and playout controls. The last paper was on *Agent-Oriented Knowledge-Based Distributed Economy* by Professor P. Gmytrasiewicz, University of Texas, Arlington. He addressed the issue of evaluating the value of information and its presentation from the points of : the consumers of information and producers of information.

2 Overview of Working Group Discussions

At the beginning of the workshop, small groups of 3-5 researchers were formed in the areas of *Networks/Operating Systems*, *Multimedia databases*, and *Intelligent Reasoning*. The charter of these working groups were to create summaries of important research directions in each of the above areas. The chairs of each of the 3 groups met in order to study the relationships across the different areas listed above.

The working group on *Intelligent Reasoning* was chaired by Dr. Lokendra Shastri with M. Herzog, P. Gmytrasiewicz, A. Nerode, and A. Rajashekar as constituent members. The main issues identified as part of the *Intelligent Reasoning in Multimedia Databases* were:

- Content-Based Query Transformation
- Content-Based Navigation and Matching
- Content Extraction/Annotation
- Efficiency: Real-Time and Off-Line

- Content Creation
- Data Mining

The second working group on Multimedia Databases was chaired by Professor Sushil Jajodia with L. Golubchik, B. Prabhakaran, C. Yu, S. Balachandar, E. Hwang and W. Aref as constituents. In this working group, research issues in data acquisition, data modeling, query language, query processing, indexing and distributed heterogeneous sources of information were discussed. The group saw the need for preprocessing the multimedia information by generating annotations (automatic/semiautomatic), storing the annotations along with the raw data and integration of annotations from different media streams. For multimedia query processing the working group discussed the need for incorporation of approximate matching along with interactive and probabilistic retrieval of information. The working group also discussed the issues in indexing different media of information such as the many dimensions involved (temporal, spatial, etc).

The working group on *Networks/OS For Multimedia* was chaired by R. Muntz, and consisted of S.K. Tripathi, P. Venkat Rangan, H. Kanakia, P. Mishra. The working group identified the need for translating end-to-end QoS to hop-by-hop QoS requirements that need to be available or enforced. The task of determining the QoS requirements for a multimedia application was also emphasized. The group also discussed the pros and cons of rate-based and delay-based resource reservation schemes. For the OS support, the group identified the necessity for restructuring I/O systems for multimedia applications. They also discussed whether there is a need for providing uniform I/O abstraction. Security issues, especially the need for an end-to-end view of security, were also discussed.

3 Workshop Publication

Attached to this report is a copy of the preprints of the workshop proceedings.



PRE-PRINTS

First International Workshop on Multimedia Information Systems

**Doubletree Hotel National Airport
Arlington, Virginia
September 28-30, 1995**

**Chairs:
V. S. Subrahmanian and
S. K. Tripathi**

**Sponsored by:
Army Research Office and
University of Maryland Institute for Advanced Computer Studies**

Table of Contents

The MPEG Multimedia Stream Architecture	1
<i>P. Venkat Rangan, University of California at San Diego</i>	
Virtual World Data Server	3
<i>Richard Muntz, William Jepson, Walter Gekelman, Walter Karplus and D. Stott Parker, University of California, Los Angeles</i>	
Multimedia Applications in an IP over ATM Environment	12
<i>Dilip D. Kandlur, IBM T. J. Watson Research Center</i>	
Multimedia Information Systems in Open World Domains	13
<i>Marcus Herzog, Vienna University of Technology and Paolo Petta, Austrian Institute for Artificial Intelligence</i>	
Network Protocols for Wireless Multimedia Access	17
<i>Partho P. Mishra and Mani B. Srivastava, AT&T Bell Laboratories</i>	
Multimedia Networking	25
<i>Jim Kurose, University of Massachusetts</i>	
New Languages for the Integration of Heterogeneous Data Sources	26
<i>Peter Buneman, University of Pennsylvania</i>	
Foundations of Multimedia Information Systems	28
<i>Sherry Marcus, 21st Century Technologies, Inc. and V. S. Subrahmanian, University of Maryland</i>	
RoadRunner: An Operating System for Multimedia Applications	29
<i>H. Kanakia, AT&T Bell Laboratories, D. Saha and S. Tripathi, University of Maryland</i>	
Indexing Multimedia Databases	30
<i>Christos Faloutsos, University of Maryland and AT&T Bell Laboratories</i>	
Adaptive Query Reformulation in Content-based Image Retrieval	39
<i>Gwang S. Jung, Jackson State University and Venkat N. Gudivada, Ohio University</i>	
Research Into Multimedia Database Systems	50
<i>Jeffrey D. Ullman, Stanford University</i>	
Query Refinement in Constraint Multimedia Databases	51
<i>Elisa Bertino and Barbara Catania, Università degli Studi di Milano</i>	

Advanced Video Information System: Data Structures and Query Processing	60
<i>Sibel Adali, Kasim S. Candan, Su-Shing Chen, Kutluhan Erol and V. S. Subrahmanian, University of Maryland</i>	
Hybrid System Methods for Distributed Multimedia Systems	61
<i>Wolf Kohn, Sagent Corporation, Anil Nerode, Mathematical Sciences Institute and John James, Sagent Corporation</i>	
Benchmarking Digital Video	68
<i>Richard Gerber, University of Maryland</i>	
Multimedia Information Systems	69
<i>Sherry Marcus, 21st Century Technologies, Inc.</i>	
Integrating a Scalable Rapid Reasoning System with a Database System	71
<i>Lokendra Shastri, International Computer Science Institute</i>	
On Storage and Retrieval of Generalized Spatial Data	79
<i>Naphtali Rishe, Florida International University</i>	
Content-based Retrieval of Pictures and Videos	80
<i>A. P. Sistla and Clement Yu, University of Illinois at Chicago</i>	
Multimedia Information Systems	89
<i>Bob Allen, Bellcore</i>	
Managing End-system Resources for Predictable Quality-of-service	90
<i>Raj Yavatkar, University of Kentucky</i>	
A Framework for Conceptualizing Structured Video	95
<i>Martha L. Escobar-Molano and Shahram Ghandeharizadeh, University of Southern California</i>	
Object-oriented Modeling and Querying of Multimedia Data	111
<i>Arif Ghafoor and Young Francis Day, Purdue University</i>	
Agent-Oriented Knowledge-based Distributed Economy	120
<i>Piotr J. Gmytrasiewicz, University of Texas at Arlington</i>	

The MPEG Multimedia Stream Architecture

P. Venkat Rangan
Multimedia Laboratory
Computer Science Engineering Dept.
University of California at San Diego
La Jolla, CA 92093-0114
E-mail: venkat@chinmaya.ucsd.edu, Phone: 619-534-5419

Abstract

Multimedia services, in which stored media objects are retrieved on demand by end users, are rapidly emerging to be offered on digital communication networks. There are two distinguishing requirements of media playback: intra-media continuity and inter-media synchronization. In the emerging international multimedia encoding standard MPEG (Motion Picture Experts Group), continuity and synchronization are handled at different layers of the multimedia stream. In this talk, we will present the MPEG standard, discuss how continuity and synchronization are specified within the MPEG standard, and propose techniques for their implementation within a distributed multimedia environment.

In order to understand MPEG synchronization, it is necessary first to familiarize with the encoder's system architecture, without of course going into details of the encoding algorithms themselves. At the time of encoding in MPEG, in the first stage, video frames and audio samples are separately encoded, and an independent stream of bytes is output for each media channel. Each media stream is then packetized independently. Packets from the different streams are interspersed to form a single multiplexed stream of packets, and the multiplexed stream is then organized into packs, with each pack comprising of an integral number of packets. The multiplexed stream is called an ISO 11172 stream.

At the time of decoding, the packs are broken into packets, and packets of different media streams are demultiplexed into their respective decoders. The video decoder then outputs its presentation units, which are frames, and the audio decoder outputs PCM samples. The demultiplexor and the media decoders are together referred to as the System Target Decoder (STD).

In a MPEG stream, whereas continuity is handled at the pack layer, synchronization is handled at the packet layer. We first present the pack and packet architectures, and then address continuity and synchronization. We will carry out all of the discussion in the context of MPEG-I media streams, mainly because, our experimental system is implemented using real-time MPEG-I hardware, with which we have direct practical experience; the more recent MPEG-II standard does not yet have commonly available hardware platforms. However, all of the main principles of media synchronization that we develop apply equally well to MPEG-II streams, whose main distinction from MPEG-I is in the compression layer; the only changes may be in the header formats of packs and packets in the media streams.

Virtual World Data Server

Richard Muntz
William Jepson
Walter Gekelman
Walter Karplus
D.Stott Parker

University of California, Los Angeles
Los Angeles, CA 90095

Abstract

During the past decade, multimedia information systems have emerged as an essential component of many application domains. Video conference and video-on-demand systems are the most prevalent and are making a transition into more wide spread use. Three-dimensional, interactive worlds are the next generation of multimedia systems. These worlds can be realistic representations of cities or imaginary worlds with terrains representing electric and magnetic fields invisible to the eye. Continuing technological advances in processor power, storage technology and networking will soon enable a qualitatively different functionality in multimedia systems.

We are engaged in the design and building of a high performance real-time virtual world data server. This real-time database system will be designed to efficiently store and retrieve large amounts ($> 1TB$) of spatially distributed heterogeneous data. The server will be designed to provide service for multiple concurrent real-time 3D interactive sessions. The requirements for this server are derived directly from actual experience which UCLA has acquired in creating both technology and applied demonstration projects using its own three-dimensional real-time simulation and visualization systems. These projects which include urban simulation, medical applications, and scientific visualization, are currently limited by the lack of data management support; all of these applications are limited in scope and size because all data has to fit in main memory. The urban simulation system for example, requires approximately 2.5 MB per city block so that a 100 MB main memory limits a realtime session to 40 city blocks. Based on this experience and an estimation of the size of developed areas, extension of the urban simulation to cover the entire LA basin is estimated to require 50 GB in geometry models and over 1 TB in image textures.

We will develop the underlying data management technology, which the on-going "real world" projects in urban design and planning, scientific visualization, and medicine have shown is needed. We have deliberately chosen several very different state-of-the-art applications which have been developed at UCLA to drive the design of the Virtual World Server. While different, these applications have similarities in their database and visualization requirements. In addition some advanced applications will require a combination of techniques. This diversity also forces us to develop general solutions where possible and better understand where the problems are when a general solution is not feasible. While functionally rich, these applications are all currently limited in scale due to storage server limitations. Additional functionality as well as orders of magnitude increases in scale are envisioned but each will require significant advances in data management.

Herein we further describe these applications, the problems they face and propose an integrated approach to solving the problems.

We expect to provide the solution for an increase of two orders of magnitude in the size of the problem that can be handled on a single workstation platform by extending the database to a disk based storage server but retaining real-time service. We expect to provide for an additional three orders of magnitude in the size of the problem that can be handled and to a multiuser environment by extending the system to a scalable, shared-nothing server. In addition we intend to provide for a much richer functionality through extension to interaction with external spatial databases.

1 Introduction

1.1 Problem Statement

After a decade of evolution of multimedia systems, spatial DBMS, visualization tools, virtual reality applications, and problem solving environments, a new generation of information systems is emerging. Advances in processor, display, and storage technology now make it possible to combine aspects of these various systems into information managers that, for the first time, truly exploit human information processing abilities. For example, it is now possible to render photorealistic 3D interactive animations in real-time. A great deal rides on understanding what sorts of information systems can be built to take advantage of this qualitatively different world.

We believe one type of system that will prevail is the *virtual world server*. By this we mean an information manager that allows user clients to navigate vast 3D or 4D data bases, and query associated information, in real time.

Such a system is in use today at UCLA, allowing interactive exploration of Los Angeles. With it one can fly (or drive) through various neighborhoods, inspect realistic architectural models of buildings, make queries about aspects of the buildings, etc. The system has a number of uses in urban planning: it is aiding in the rebuilding of earthquake, fire, riot and flood damaged areas of Los Angeles (and soon Kobe). It allows community members from all walks of life to directly participate in the planning process. It is generally of use in education, emergency response, health care delivery, environmental research, and civil infrastructure.

Virtual world servers have significant applications in science, medicine, and other areas such as manufacturing. They will support the new movement from visualization to VR in computational science, permitting scientists to navigate through massive amounts of data in ways never before imagined. Medical research and health care will also certainly benefit from the ability to explore better models interactively. Generally it seems virtual world servers will become an enabling technology for many activities, ranging from distance learning to collaborative problem solving, in much the way that databases have.

These applications all share two significant characteristics:

1. *vast scale*, easily requiring management of terabytes of information. The scale clearly suggests that client-server architectures will succeed, and will replace the in-memory, single-user workstation architectures used today.
2. *real-time interaction*, with 'realistic' movement in the virtual world. In applications involving time there is also a rate at which simulated time is advancing in the virtual world. That rate may be adjustable by the user but, between such adjustments, should proceed, in real time, at a constant rate. Interruptions or variations in this rate that are perceptible to the user can destroy the illusion of the virtual world and greatly diminish or completely eliminate the system's usefulness.

These characteristics are seemingly at odds. How can one build a server supporting both this vast scale and real-time interaction?

1.2 Basic Approach

The problem is to generate high-quality animation at a given frame rate with the viewpoint being controlled by the user. (The user also has other controls but the main one is being able to move about in the virtual world.) At the same time, virtual world servers are database managers. A major component of virtual environments is the database that represents a model of that world. That model contains representations of the objects of the world and their behaviors.

Our approach is to develop a real-time storage server (RTSS) that marries real-time media handling with database management, drawing on our experience in both areas. To test our design, we will experiment with use of the RTSS on several state-of-the-art applications, interacting directly with domain experts.

The applications will fall into two main categories. The 3D model based simulations, exemplified by the urban simulation, requires storage and retrieval of geometric models (e.g., of buildings) and image textures (e.g., the sides of buildings, etc.). Here the geometry and texture is stored on disk and the problem is timely delivery of exactly what is needed to the rendering engine. For the scientific and medical visualization applications the data stored on disk will consist mainly of 4D arrays of scalars, vectors, tensors, etc. The processing pipeline in this case is somewhat different. The disk-based storage system must fetch the desired "raw" data from the disk at the required rate, and visuals (e.g., vector fields, isosurfaces, etc.) are generated "on the fly" and sent to the rendering engine.

1.3 Challenging Technical Issues

Until recently "real-time disk-based storage system" would have been considered an oxymoron by many people. A major problem is that disk efficiency (throughput) and request latency are very much a function of the scheduling algorithm due to the rather significant seek and rotational latency times. Thus a disk driver had to be free to reorder requests to make the most efficient use of the device, and this is inconsistent with tight real-time constraints associated with the individual requests. However recent work on video-on-demand servers (e.g., [10]) has demonstrated that efficient utilization of disks is possible in conjunction with guaranteed real-time delivery of continuous media (video, audio). In these systems a user may have to wait before their request can begin but, once begun, it is guaranteed that the display (a) will not 'starve' and (b) will not have to buffer more than a specified amount of data.

The virtual world server problem is even more technically difficult to solve than problems presented by media servers because of (a) the more complex nature of the data, and (b) the interactive nature of the application. Note that for video-on-demand the only interactions that are generally allowed are VCR functions such as fast forward, pause, etc. and these are not high frequency events. However, data rates in many of the virtual reality applications will be lower than with video-on-demand. In the virtual worlds we are often dealing with models and video frames generated from the model as projections. Therefore the same model data is used repetitively on successive frames.

Most work on 3D interactive graphics has assumed the display database is main memory resident. There are two notable exceptions of which we are aware. The first is the work on interactive architectural walk-throughs [15, 4, 5] in which a disk based display database was implemented. This work incorporates the basic concepts of a real-time storage hierarchy but is a single user system and on a small scale. It did not consider multiple concurrent types of workload, combined display of 3D geometric models and array data, data integration with external spatial databases, and many other aspects of the problem that will be included in our system. The second related work is a project at SRI to provide 3D interactive access to a landscape constructed from elevation data and aerial images [13]. This system deals with large data sets and uses levels of detail to advantage in reducing the workload. The Terravision project addresses some of the issues but, again, in a more limited context than we plan to address.

High utilization of resources is desirable to achieve a good cost/performance ratio. However, high utilization is inconsistent with guaranteed real-time service and statistically varying service requirements. There

are two main aspects to the variation in service requests. One is the variation in arrivals of new users to start a "session" in the virtual world. As in the video-on-demand case, a user may have to wait until resources are available. The system should, however, guarantee a certain "quality of service" once the session is started. The user, in turn, must usually promise to limit his or her behavior (e.g., to a certain speed of motion) in order to get this guarantee. Within the bounds agreed to, the user's resource requirements will still be time-varying over a wide range. Without reservation of the maximum resources needed for each active session, the system cannot deterministically guarantee a fixed level of service. Therefore most systems provide a "statistical guarantee", e.g., delay jitter will be less than 5 msec. for 99.5% of all frames.

In some real-time systems the quality of service is easy to quantify (e.g., the error rate or the frequency of buffer overruns). In dealing with human perception (as in the 3D interactive graphics case) the quality of service is harder to define, as it is related to the perception of the user. This leads to complex tradeoffs in terms of "real-time" service. Clearly at a time when "full quality" cannot be delivered to each active session, a compromise can be made in which there is some degradation of service to some active sessions. The major methods for compromising service quality will be (1) frame rate and (2) level of detail. The former must be done very sparingly for many of the interactive applications. The latter is more complex because (a) there are a number of different approaches to achieving it and (b) there are multiple factors that determine what is "acceptable". We will develop as part of this project, definitions of quality of service appropriate for virtual world environments, methods of specifying how tradeoffs are to be made, and system algorithms for implementation of these policies.

A final challenge comes from the need to integrate multiple forms of information. First, it is inevitable that video playback will be also required from the server. In the applications described below, it is also important, however, to be able to relate a virtual world with one or more databases. For example, with urban simulations it is important to be able to be "instantly transported" to a particular building, street, or river, find all buildings owned by a particular person, find neighborhoods satisfying demographic queries against census data, etc. Thus the virtual world server must be able to integrate the virtual world concept with various databases to provide a rich and expressive query/response interface.

2 Driving Applications

We have deliberately chosen several very different applications to be "functionality drivers" for the virtual world server. While functionally rich, these applications are all currently restricted in scale due to storage limitations. Additional functionality as well as orders of magnitude increases in scale are envisioned.

2.1 Real-time Urban Simulation

Drawing on technologies developed for virtual reality and military real-time simulation, researchers at UCLA have created a computing environment designed for real-time urban simulation. The UCLA Urban Simulation System supports 3D interactive "fly through" of areas for which models have been constructed. An unusual feature of this system is that the visualization is photorealistic. A combination of aerial photographs, digital elevation maps, and video imagery from a site survey are combined to build the models. The system support "magic carpet" transportation in that a user can fly over the city or drop down and drive along the streets in a smooth, natural fashion.

We are currently working directly with numerous municipal entities in further developing and deploying this technology. We feel, as do they, that this application has the promise of dramatically effecting the day-to-day quality of life for virtually every city inhabitant. It has been shown to be equally effective in affluent and traditionally under-served communities. Even in these difficult economic times a number of cities have already provided funding to UCLA for demonstration projects designed to introduce the systems of the future

to the planners of today. These cities and municipalities are eagerly awaiting the day (in the not too distant future) when this system will maintain a model of the entire Los Angeles (or San Francisco) region for remote access and collaborative regional problem solving. This data will be distributed via a high speed, wide-area ATM network currently being installed and will be used for a wide variety of diverse applications including: emergency response and crisis management; community involvement in planning decision making; regional traffic and congestion management; air and water quality and other environmental monitoring activities; distance learning; driving simulation; collaborative problem solving and numerous others.

The urban simulation system is already being employed in several cutting edge urban redevelopment projects. These projects will be used to provide content for the high-performance data-base. The system has been designed to allow members of affected communities to participate in the redesign and rebuilding of their communities via the use of advanced technology previously reserved to the military. Additionally, in response to requests from Japan, we have agreed to provide the system to the city of Kobe to aid in their rebuilding efforts.

The data delivery problems in Urban Planning are formidable. For example, a typical square mile of urban real estate contains between 150 and 200 blocks. Our current models typically require between 100 and 200 images and between 1,000 and 10,000 polygons to fully represent each block. Our objective is to provide a structure which will allow the creation of a database sufficiently scalable to contain the greater Los Angeles region (4000 sq. mi.) and eventually California, while maintaining real-time performance. If we were to model just the populated areas of the region (approximately 1200 square miles) we would have approximately 200,000 city blocks, 4,000,000 buildings, 30,000,000 individual images and 500,000,000 polygons. Our data storage requirements would be approximately 50GB for the geometry and 1TB for the image data.

2.2 Virtual Aneurysms

In another ongoing project, clinicians at the UCLA School of Medicine are collaborating with a team from the Computer Science Department on a highly-innovative project that appears destined to play a crucial role in the treatment of brain aneurysms. Aneurysms are potentially fatal dilations (bubbles) on the walls of blood vessels, and are normally mitigated by surgery. This becomes impossible when the aneurysms are located deep in the brain and therefore inaccessible. To cope with this threat, that annually takes tens of thousands of lives, UCLA radiologists have invented a non-surgical and very effective procedure to maneuver (via a major artery) a coil of platinum wire into the mouth of the aneurysm, so as to plug it and thereby render it harmless. Versions of this technique have already been used on over 2000 patients.

In order to avoid rupturing the aneurysm, the physicians need to have detailed information regarding the pressures and shear forces within the aneurysm as the heart goes through its pumping cycle, as well as how these can be expected to change as the coil is inserted. This is where the computer science team enters the picture. Using a virtual reality visualization, the physician is afforded the opportunity of 'taking a walk' through the aneurysm and can observe the transient fluid flow (hemodynamic) variables, prior to the procedure, at various stages of coil insertion. There have been numerous applications of virtual reality to medicine, but the UCLA method is unique in that it superimposes the outputs of computational fluid dynamics (CFD) simulations upon the geometry of the aneurysm and nearby blood vessels as revealed by MRI and CTR images. (As the size of our data sets grow, a storage server supporting both types of data, such as proposed here, is precisely what will be required.) Very significant in its own right, this approach has important applications in a number of other problems in clinical medicine.

The data set required for one session in which a physician examines a virtual aneurysm requires approximately 2 GB of output data from the simulation. These data are converted "on-the-fly" to visuals to represent isosurfaces, vector fields, etc. There are several potential bottlenecks: access to the array data pertinent to the computation of the visuals for the next frame, computation of the visuals and finally, actual

rendering of the visuals. To achieve a tolerable frame rate of 20 f/s, all results must be computed on the order of 20 ms. Typically one cycle may require computation of 1000 or more individual data points. This is another good application for parallel processing, as each data point can be computed independently.

A reasonable goal is to be able to render in stereo at 10-30 frames/sec. One frame will have approx. 20-40,000 polygons. Thus we need actual render rates of: $40,000 \text{ polygons/image} * 2 \text{ images/frame} * 15 \text{ frames/sec} = 1.2 \text{ million polygons/sec}$. This render speed requirement will increase as the visualization system becomes more sophisticated. Device (e.g. trackers, gloves) latencies should be below 50 ms.

2.3 Plasma Physics

A plasma is a medium which exhibits complex nonlinear and often non-local behavior. These effects, which are fascinating in their own right also occur in space and fusion devices. Advances in the design of plasma sources, detectors and diagnostics have made possible the detailed study of many of these effects in terrestrial laboratories. One such device, the Large Plasma Device (LAPD) [6], has been constructed recently at UCLA. The highly reproducible and quiescent plasma it produces allows for great flexibility in design of experiments. The LAPD is the premier research tool for the detailed study, in the laboratory, of processes that occur in space. Aside from the device the laboratory is equipped with a state-of-the-art data acquisition system [7]. Since its construction, key experiments in the interaction of current systems [8], Alfvén waves [9], and the interaction of whistler waves with non-uniform plasmas [1] have been performed.

A team of scientists from UCLA and several other institutions are presently engaged in numerous experiments using the LAPD. This follows the current trend in science, which is toward medium-scale experiments shared by several research groups where an experimental device is too large and expensive to be replicated at other institutions. Data from such experiments must be offered to all collaborators over fast (ATM) networks. Data can be downloaded or accessed through remote compute servers for visualization and analysis. The implications for distance learning of science are also clear.

The data sets acquired in these experiments are enormous, which makes their analysis cumbersome. For example, data from a typical experiment would include measurements of a distribution function, electric field, magnetic field, scalar fields such as density and temperature. Electric and magnetic field data (both vectors) is acquired at approximately 10,000 positions, and 100,000 temporal values (9 GB). A data set, (or a set of initial conditions) of this size presently takes about a week to acquire but will soon take only hours.

For example we have rendered a cross sectional view of the magnets surrounding the device. Also shown is the vacuum wall, the plasma source and an antenna which launches Alfvén waves. The waves magnetic field is rendered as one isosurface which is hollow. Also the vector magnetic field of the wave are visualized as glyphs..

In this application area the problem is obviously the magnitude of the data sets and the real time requirements for delivery to the graphics rendering subsystem. In fact, the data bandwidth requirements for scientific visualization application will generally exceed the bandwidth requirements for the urban simulation type applications. Advances in probe manufacture and parallel acquisition will put more pressure on analysis capabilities. We expect that within ten years, with the introduction of microscopic detectors and massively parallel acquisition systems the acquisition time could drop to seconds. Aside from data acquisition, issues such as management and analysis are paramount.

2.4 Characteristics of the Applications

With geometric models, most of the information (geometry and image texture) are known a priori. Example applications are: urban simulation, architectural walkthroughs, emergency response, etc. These applications are characterized by (but are not limited to) a fixed spatial distribution of objects. The *primary* source of animation in the viewed scene is the change of perspective of the user as the he/she moves through the

virtual world. The geometric model of the static portion of the world is stored on disk and must be moved into the main memory of the rendering engine such that the visible portion of the world is resident. The data rates are (on the average) modest for a single session (estimated at about 5-10 MB/sec. for the current Urban Simulation application). Culling and systematic use of controlled levels of detail are important to (a) control quality of the display at time when the I/O bandwidth peaks and cannot be met by the storage server and (b) more generally to keep from saturating the rendering engine.

Volume data, as in scientific 3D and 4D (= temporal 3D) visualization, present very different requirements. For one thing, the visual representation is a (to some extent ad hoc) *model*: not a natural rendition of what a user would see in real life, but a visual representation. With 4D data the most important difference is that the content of successive frames is not often computed (by projections) from geometry and image textures which remain static but, rather, each frame must be recomputed from the "raw" data (3D spatial data plus time). Volume data arrays are either measured or generated by a computational model. In general these data sets will be pre-generated and stored for later access. This is necessary since these scientific models are often computationally complex, and measured data are often not sampled at a rate suitable for real-time display (e.g., the time scales are very different).

Two strategies are possible in generating 3D visuals (vector fields, isosurfaces, streamlines, etc.) from raw volume data: on-the-fly generation, or offline generation followed by interactive visualization of the pre-generated set. A simple calculation shows that in reasonable 4D applications, pre-generation of the geometry and texture for each time step, in even a moderately sized data set, will either generate a prohibitively large set of data or will be unnecessarily rigid in representation. (Note that unlike the model-based applications, here the geometry and texture is changing at each time step.) Thus the second approach is necessary. This implies that the data stored "permanently" on disk is the 4D array data, and the pipeline to the display must create the geometry and texture for each frame. This can be a computationally expensive step and is a potential bottleneck. This approach, however, requires that the science-minded groups have sufficient computing capacity to do this.

Virtual worlds consisting of geometric model and volume data therefore differ in several significant ways:

- *storage structures*

Geometric models are stored as polygons, attribute data and image texture, while volume data are typically just arrays.

- *computational requirements*

Between the storage server and rendering engine, geometric models require very little computation, while volume data have potentially high computational requirements.

- *I/O requirements*

Geometric models are likely to be less I/O intensive for individual users. However, concurrent multiple users of the storage server will generate stress on resources even for the model-based applications.

With both approaches, culling and strong use of levels of detail are needed to deal effectively with transient saturation of resources. (The assumption is that we can not design a system to simultaneously use resources efficiently, i.e., maintain high utilization, and give deterministically guaranteed service. Therefore we will provide high utilization with rules for back off from high levels of quality when resources occasionally are saturated.)

At this point the concepts of culling and controlled level of detail have been used extensively in rendering geometric models. In volume data rendering, 'culling' and 'level of detail' can possibly take on mathematical meaning, with for example reduced precision or grid step sizes allowing faster computations. It is an interesting and open issue how much "commonality" there is for these concepts between geometric model and volume data. In models that can be related precisely to volume data, a great deal of commonality seems

likely. For instance, it seems possible that culling and level of detail can be adapted naturally for isosurfaces and vector fields.

The geometric model and volume data applications are certainly not disjoint. In the urban simulation model, for example, we plan to incorporate CFD simulation data for wind pressure analysis within a 3D urban model. We could, for example, render a vector field visualization of wind around a skyscraper. Additionally, the automated creation of large digital terrain models (DTM's), replete with multiple precomputed levels-of-detail, generated from remote sensed USGS Digital Elevation Map (DEM) data will, in general, echo the required preprocessing of the scientific data.

In fact, the applications are also similar in their requirement for integrating the virtual world concept with various databases, to provide a richer, more expressive query/response interface. For example, consider the following scenarios:

- A user, via a query form interface, asks for "All liquor stores that are within a quarter mile of an elementary school in the LA basin." The system responds with an aerial view of the LA basin with pinpoints of light where the liquor stores are. By clicking on one of the lights, the user is immediately positioned in the street facing the liquor store and from there can move about the neighborhood. Note that the initial query response is not real-time but once the user is "transported" to the neighborhood, the nature of the interaction reverts to real-time.
- A user at a PC or small workstation asks for directions to go from UCLA to the Santa Monica Pier from a WWW server on our system. A spatial database is used, with an existing application program, to select an appropriate route. The 3D model of the route along with a specification of that route is returned in VRML (Virtual Reality Modeling Language) [17] and a local viewer allows the user to view the route in 3D. The user may be able to have limited motion ability along the path or limited distances from the path. From the point of view of the data server this is a completely non real-time application. It becomes real-time for the viewer on the user's computer.
- An automobile engine designer builds a model of a new engine with a novel exhaust valve design. He/she runs a simulation and collects data on temperature, pressure, etc. during the four engine cycles. Combining the 3D geometric model and the simulation output the designer can visualize the combustion pattern. In addition, to compare the design with several different parameter settings, two or more synchronized displays may be generated. One may want to have one tracking system that feeds each display so that the displays are not only synchronized in time but also provide the same viewpoint.

Each example shows the integration of virtual worlds with other databases, which are related through some notion of 'location'. The final example illustrates the need for multiple virtual world displays with several options for the manner in which they are synchronized.

References

- [1] J. Bamber, W. Gekelman, J. Maggs. Observation of Whistler Wave Mode Conversion to Lower Hybrid Waves at a Density Striation. *Physical Review Letters* 73, 2990-2993 1994
- [2] S. Berson, S. Ghandeharizadeh, R.R. Muntz, and X. Ju, "Staggered Striping in Multimedia Information Systems", *Proc. 1994 ACM SIGMOD*, May 1994.
- [3] S. Berson, L. Golubchik, R. R. Muntz, "Fault-Tolerant Design of Multimedia Servers", *Proc. 1995 ACM SIGMOD*, May 1995.

- [4] T. Funkhouser and C. H. Sequin and S. Teller. Management of Large Amounts of Data in Interactive Building Walkthroughs. ACM SIGGRAPH Proc. of the 1992 Symposium on Interactive 3D Graphics. 1992.
- [5] T. Funkhouser and C. H. Sequin. Adaptive Display Algorithms for Interactive Frame Rates During Visualization of Complex Virtual Environments. Proc. SIGGRAPH. 1993.
- [6] W. Gekelman, H. Pfister, Z. Lucky, J. Bamber, D. Leneman, J. Maggs. Design, Construction, and Properties of the Large Plasma Research Device - The LAPD at UCLA. *Rev. Scientific Instr.* ,20, 614-621 (1993)
- [7] W. Gekelman. Data Acquisition Systems. *Encyclopedia of Applied Physics* Vol 4, 463-483, VCH Publishing Co, (1992)
- [8] W. Gekelman, J. Maggs, H. Pfister. Experiments on the Interaction of Current Channels in a Force Free Plasma-Relaxation to a Force Free State. *IEEE Transactions on Plasma Science* , 20, 614-621 (1993)
- [9] W. Gekelman, D. Leneman, J. Maggs. Experimental Observation of Alfvén Wave Cones. *Physics of Plasmas* ,1, 3775-3783 (1994)
- [10] D.J. Gemmell, H.M. Vin, D.P. Kandler, P.V. Rangan and L. Rowe *Multimedia Storage Servers: A Tutorial*, IEEE Computer, May 1995, pp. 40-51.
- [11] L. Golubchik, J.C.S. Lui, R.R. Muntz, "Reducing I/O Demand in Video-on-Demand Storage Servers", *Proc. ACM SIGMETRICS '95*, to appear.
- [12] R. Liggett and W. Jepson. Real-time visual simulation technology for urban planning/design decision making. *Proceedings of the Fourth International Conference on Computers in Urban Planning and Management*. Melbourne, Australia July 1995.
- [13] Y.G. Leclerc and S.Q. Lau, "Terravision: A Terrain Visualization System", *SRI Technical Note No. 540*, April 1994.
- [14] R. Liggett and W. Jepson. Use of Real-time Visual Simulation Technology for Urban Planning/Design Decision Making. *Proceedings of the Fourth International Conference on Computers in Urban Planning and Management*. July, 1995.
- [15] S. Teller and C. Sequin. Visibility Preprocessing for Interactive Walkthroughs. *Proceedings SIGGRAPH '91*. Las Vegas, Aug., 1991.
- [16] N.M. Thalmann and D. Thalmann, Eds. *Virtual Worlds and Multimedia*. Wiley, 1993.
- [17] G. Bell and A. Parisi and M. Pesce. The Virtual Reality Modeling Language - Version 1.0 Specification. WWW document: <http://www.eit.com/vrml/vrmlspec.html>. 1995. Further information on VRML is available at <http://www.w3.org/hypertext/WWW/MarkUp/VRML/>.
- [18] H. Tokuda, T. Kitayama, "Dynamic QOS Control based on Real-Time Threads", *Network and Operating System Support for Digital Audio and Video (NOSSDAV'93)*, Springer LNCS # 846, pp. 114-123, 1993.

Multimedia Applications in an IP over ATM Environment

Dilip D. Kandlur
IBM T. J. Watson Research Center
30 Saw Mill River Road
Hawthorne, NY 10532
kandlur@watson.ibm.com

Abstract

It is advantageous to support multimedia applications in a IP environment since this environment is well developed and provides many important facilities such as universal naming and addressing mechanisms and interoperation over different network types. On the other hand, ATM networks provide facilities for resource reservation, delay guarantees, and high bandwidth. Hence, it is prudent to consider supporting multimedia applications in an IP over ATM environment.

We examine two key aspects of the ATM deployment in the TCP/IP environment – Logical IP Subnetworks (LISs) and Address Mapping Servers, with a specific emphasis on how different alternatives for LISs and Address Mapping Servers enable efficient exploration of all the capabilities that ATM promises to deliver. We propose an approach for ATM deployment in the TCP/IP environment that leverages the existing technologies (e.g. DNS, routing protocols), minimizes new development, imposes no new requirements on ATM, allows for small, incremental changes with immediate payoffs, while at the same time retains backward compatibility with the existing schemes, such as LAN Emulation and “classical” IP over ATM. The proposed approach combines switch-based infrastructure with router-based overlay and uses each for that which it is best suited: switch-based infrastructure for applications that can justify an SVC establishment; router-based overlay for all other applications.

[Joint work with Yakov Rekhter]

Multimedia Information Systems in Open World Domains (Extended Abstract)

Marcus Herzog
CD-Lab for Expert Systems
Vienna University of Technology
A - 1040 Vienna, Austria
Email: herzog@dbai.tuwien.ac.at

Paolo Petta
Austrian Research Institute for
Artificial Intelligence*
A - 1010 Vienna, Austria
Email: paolo@ai.univie.ac.at

1 Introduction

Computer-based information systems manage resources which represent properties of the covered domains. Typically, human decision-makers use these abstractions as foundations for their decision-making processes. Additionally, information systems may apply algorithms using domain knowledge to generate explicit representations of information that is hidden in the mass of data.

Closed world domains can be modeled with a limited number of features giving a sufficient coverage of the domain: all relevant data are collected and all knowledge needed to process these data independently of the human user is formalized. Within a closed world domain, an information system should always be capable of reaching a meaningful goal state, provided that the end user operates within the limits of the modeled domain. Successful application examples can be found e.g. in traditional expert systems.

In contrast, for information systems in *open world domains* it cannot be assumed that all possible states of the world are covered. The semantic dimensions of such domains are not limited and thus cannot be fixed *a priori*. This necessarily incomplete coverage of the domain will lead to so-called *breakdowns* ([5], or see also e.g. [1]), that signal the transition from the currently covered part of the domain to territory that has not yet been formalized. In the event of a breakdown, processing within the information system cannot be continued without intervention by

*Supported by the Austrian Federal Ministry for Science, Research and The Arts under grant GZ 607.515/3-II/6/94

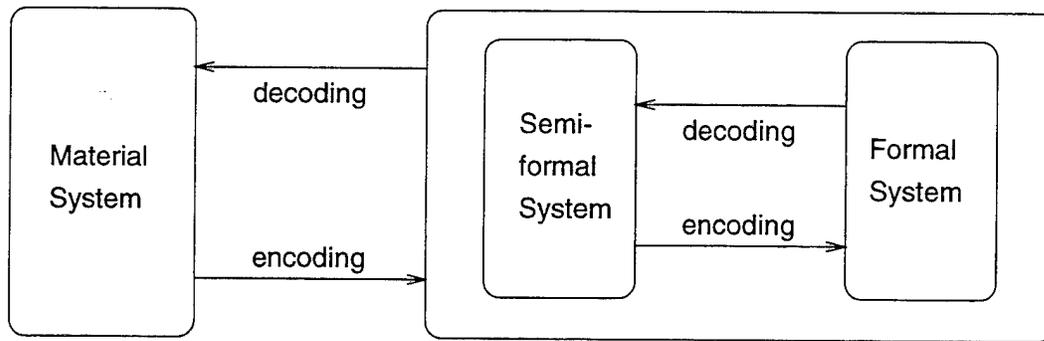


Figure 1: The acquisition process

human operators who have to either redefine the query or modify the contents of the knowledge base.

In the following we will argue for the use of multimedia components within information systems to provide guidance in handling such exceptions. We will present an information system model for open world domains and show a prospective application area.

2 A model of a multimedia information system

Our model of a multimedia information system comprises two layers: the semi-formal *hypermedia/multimedia* layer and the formal *access* layer. These two layers are used in a two-stage approach to bring objects and relations of the real world (the material system) in congruence with implications expressed in a suitable formalism (the formal system). Figure 1 shows the relations between these three systems.

The use of this two-stage knowledge acquisition approach alleviates the difficult process of abstracting real world artifacts. This is especially true for domains where multimedia can be used to give a more direct computer-based representation of those items: e.g., a picture is still better represented by a scanned image than by some verbal descriptions. Our approach relates to the theory of ontological design [5, 4], where the representation of artifacts is divided into an *architectonic* and a *semantic space*. The architectonic space accounts for explicitly represented properties of the artifacts while the semantic space covers the totality of all potential associations tied to elements of the multimedia database.

In contrast to more straightforward data acquisition methodologies this approach combines the simultaneous acquisition of formal and informal aspects of the domain: e.g., inserting a picture along with giving a formal description of some aspects of the content in an appropriate formalism. In terms of the example give

above, while the picture holds "indefinitely rich" semantic information (i.e., the picture can be interpreted in an unlimited number of ways by humans), the formal description will always just cover a limited amount of that semantic content.

The goal of an information system is to provide means for locating relevant information for a given problem setting. As a consequence of the possibility of the occurrence of breakdowns in open domains, such a system cannot operate autonomously at all times: overcoming breakdowns is only possible via assistance by human users, who creatively bring into play knowledge that is not yet formalized in the system. Multimedia components can be used in this context to help users refocus their information needs by leading them to semantically related places of the existing architectonic space or by hinting at promising directions of extension.

3 Possible application areas

So far we investigated the applicability of this approach in the field of design, more precisely in the domain of architectural design [2, 3]. We think that our model of a multimedia information system is particularly well-suited for this application area, as architectural design can be characterized as a *weak theory* domain, where no exact rules of how to process and apply knowledge can be given. Most of the knowledge involved in the design process can be represented best by a collection of design artifacts, which in turn can be stored as multimedia documents. These documents define a semantic space, as all kinds of associations can be established around the content of these documents. Indices that work as concurrent vocabularies to access the multimedia documents enable formal reasoning about the system's content.

Breakdowns occur when currently used vocabularies do not provide sufficient coverage or are in some other way inadequate to express the user's information needs. In this case, the human user can resume the stalled session by selecting a different point of view, represented by some other vocabularies. The selection of multimedia items referenced by items of both previously used and newly chosen indices gives the system the possibility to identify other relevant documents, that are in turn presented to the user. These items can be seen as "talking back" [1] to the users, i.e., they assist them by posing the question which hitherto neglected other relevant aspects of their problem are covered by them.

To conclude, we think that this model of a multimedia information system can be applied profitably to all application areas that cannot be adequately abstracted by means of formal expressions but rather need rich multimedial representations to capture the semantic content of the domain. This is especially true for fields where physical artifacts play an essential role, as their properties are usually impossible to formalize for all possible points of view.

4 Relevant research topics

Within our model of multimedia information systems we can identify the following research topics:

- *Sufficient size of multimedia collection*
What are the criteria for proving the sufficiency of the size of a collection of multimedia items in terms of richness and diversity to cover the semantic content of a domain?
- *Sufficient size of architectonic space*
What types of abstraction have to be supported by the system to allow meaningful creation of information?
- *Sufficient re-use of multimedia items*
To allow the system to reason autonomously, a sufficient number of items of the architectonic space have to be related to multimedia items. What is the critical value of this number and how can we ensure that this constraint is met by implementations?
- *Standardization and re-use (sharing)*
What are relevant standards for both the access and the hypermedia layer of the system that can be used for building such systems?

References

- [1] Fischer G.: Turning breakdowns into opportunities for creativity, in: Special Section: Creativity and Cognition, *Knowledge-Based Systems*, 7(4), 221–232, 1994
- [2] Herzog M.: *The Use of Intelligent Hypermedia in Architectural Design Environments — a Conceptual Framework*, Institut für Informationssysteme, Abteilung für Datenbanken und Expertensysteme, Technische Universität Wien, Diploma Thesis, 1994.
- [3] Herzog M., Petta P., Kühn C.: Retrieval as Exploration of Large Multimedia Document Bases, CD-Technical Report 95/79, Institute for Information Systems, Vienna University of Technology, 1995.
- [4] Kaplan N., Moulthrop S.: Where No Mind Has Gone Before: Ontological Design for Virtual Spaces, in: *ECHT'94 Proceedings*, ACM, New York, 206–216, 1994.
- [5] Winograd T., Flores F.: *Understanding Computers and Cognition: A New Foundation for Design*, Addison Wesley, 1985

Network Protocols for Wireless Multimedia Access

Partho P. Mishra and Mani B. Srivastava
AT&T Bell Laboratories
Murray Hill, New Jersey

1 Introduction

In the recent past, there has been an explosion of interest and activity in building multimedia systems. However, translating this interest into widespread consumer acceptance will require further improvements in networking technology to provide higher bandwidth and ubiquitous access. The physical switching and transmission capabilities required to provide transport at Gigabit speeds or higher are now largely mature. However, to allow new multimedia applications to make best use of the high bandwidth, existing network protocols need to be extended to provide Quality-of-Service (QoS) support. This is an area of active research, both in the Internet and ATM protocol communities, and practical solutions are beginning to emerge. The second problem of providing ubiquitous access is being tackled by service providers through ambitious projects to test wireless, fiber, or hybrid fiber-coax access technologies. Of these various choice, the cost advantages and the attractiveness of tetherless communication, in our opinion, make wireless access the technology of choice.

However, using wireless access technology requires network service providers to allow seamless communication over wireless and wired networks. This in turn requires the generalization and extension of the network protocols used in wired networks to handle the communication problems unique to the wireless network and to allow internetworking of the two types of networks. In this paper, we will describe an architecture for achieving this integration and provide some details of SWAN, an experimental network that we are building to test our concepts.

2 Network model

Our view of the future networked computing environment is shown in Figure 1. We visualize the use of wireless networks to provide *last hop* access to wired backbone network resources, due to the current limitations in aggregate *air bandwidth*. It is likely that the wired network will consist of a hierarchy of Local Area Networks, Metropolitan Area Networks and Wide Area Networks. We assume that the wired network architecture is based on virtual circuit based packet switching. The use of such a paradigm allows the network to more easily provide the per-connection Quality of Service (QOS) guarantees needed to support voice and video traffic. One likely realization of such an environment in the near future is via the deployment of Local Area and Wide Area ATM networks running the same signalling protocol.

We assume a micro-cellular radio-based wireless network, with cell sizes of less than 20/30 square feet. Special switches called base stations provide a gateway for communication between the wired network and the mobile hosts in a cell. Mobile hosts are allowed to roam between cells with no restrictions of any form on the mobility pattern. However, a mobile host is assumed to send and receive all its traffic through the base station in its current cell. A mobile can have all the capabilities of a regular network host and is therefore able to participate in complex signaling and data transfer protocols.

There are several open problems that need to be solved in designing network protocols that support the above model of communication:

1. How should the wired network keep track of the location of mobile hosts?
2. How should data flow over connections terminating or originating at a mobile host be maintained in the presence of host mobility?
3. How should per-connection QoS requirements be guaranteed in the wireless network?
4. What form of error and flow control should be used?
5. What Application Programming Interface (API) should the network protocols provide to applications?

We are currently addressing these questions through the design and implementation of an experimental network christened SWAN (*Seamless Wireless ATM Networking*) [1].

3 SWAN System Architecture

The SWAN network implements the network computing model depicted in Figure 1. The wired backbone network in SWAN consists of workstations and PCs interconnected via a hierarchy of FORE ATM switches. The wireless last hop consists of workstations and PCs (basestations) and portable computers (mobiles) communicating via a wireless ATM adapter card [9]. The organization of the software on the basestation and mobile host is depicted in Figure 2. In this document, our primary focus is on the design of the *data transport (DT)* and *connection manager (CM)* modules. We also describe the interaction between the CM and the MAC protocols in specifying and guaranteeing the end to end QoS requirements. We refer the reader to [2, 7, 9] for details of the lower level software and hardware components of SWAN.

The key functions implemented by the CM and DT modules, are address resolution, connection establishment and rerouting, QoS support, and reliable data delivery.

3.1 Location and Connection Establishment

SWAN uses a twin-address scheme to provide support for host mobility. Each host has a unique identifier that is used to identify it as a communication endpoint. Each host also has a location

which identifies its current point of attachment to the network - thus a mobile host's location is merely the identifier of its current base station. Location servers are used to maintain the mapping between host *identifiers* and host *locations* and are updated over time as mobiles move between cells. We remark that a location server is simply a distributed directory and conventional optimizations of caching, trading updates vs search time etc, can be used for efficient implementation of this service [6].

The protocols for connection establishment in SWAN provide support for host mobility. Both source and destination hosts are allowed to move at any time, even during the process of connection establishment itself. The first step in connection establishment involves the CM at the source host contacting the closest location server to find the location of the destination host. It then issues a connection establishment request with both the host identifier and host location fields set. Intermediate switches forward the connection establishment request based on the host location field. At the destination base station, the CM forwards the request to the mobile host using the host identifier field, if the mobile is still in that cell. Otherwise, the destination base station queries the location LS and forwards the connection establishment to the new location. The connection establishment reply message is sent back to the initiating host in an analogous fashion. These protocols are described in greater detail in [5].

3.2 Connection Rerouting

The use of end to end connections for data transfer necessitate the rerouting of these connections whenever a host moves between cells. We employ a novel solution for connection rerouting that attempts to minimize the disruption of data flow due to host mobility without causing any data resequencing. This is based on the observation that in the target environment it is likely that a mobile will frequently cross cell boundaries; however, most of the movement will be within a local domain. We argue that in this scenario, it is appropriate to perform path extensions, i.e. extend all existing VCs from the old base station to the new base station, as long as the mobile is moving in a local domain. This is beneficial because the increased cost of the network path has only a small effect on overall network efficiency since the wired LAN in a local domain typically has plenty of spare bandwidth. Also, the longer paths are unlikely to significantly affect the steady state delay and loss characteristics seen by the VCs and doing a local extension minimizes the disruption time seen by a user.

However, when a domain boundary is crossed, building an extension will cause a more costly network path to form. This may be used as a trigger to rebuild the routing tree. This rerouting strategy may be further customized for individual classes of applications, by using a specific user level performance metric to trigger the reroute. An extra optimization that may be used to improve network efficiency during the VC extension process is to detect and eliminate loops, i.e. when the same base station appears twice in the path. The protocols that implement the above model of rerouting are described in greater detail in [5].

3.3 QoS support

Over the last few years there has been extensive research in guaranteeing QoS in wired networks. However, providing QoS support for the network model depicted in Figure 1 adds additional com-

plexity. First, the air resource is shared between multiple hosts and the medium access control (MAC) protocol has to be designed to guarantee the bandwidth and latency constraints of individual connections originating/terminating at these hosts. Second, the movement of hosts between cells can affect QoS guarantees due to the time taken to do handoffs as well as the need to renegotiate for resources in a new cell.

The MAC layer in SWAN controls access to air resources by allocating *channels* to basestations and using a *token passing* mechanism to regulate access to each channel with a cell. The SWAN radios use a frequency hopping spread spectrum technique with each transmitter being required to hop pseudo-randomly among at least 75 of the 83 available 1 Mhz wide frequency slots in the 2.400 to 2.4835 MHz region, such that no more than 0.4 seconds are spent in a slot every 30 seconds. Hence a *channel* in SWAN's wireless hop corresponds to a hopping sequence, i.e. a specific permutation of the available frequency slots. In SWAN, distinct channels are defined with their own hopping sequences and distributed among the basestations in various cells. The base station controls bandwidth allocation on each channel using a token passing mechanism to grant access to each mobile. Each mobile in turn controls the allocation of air-time for each active connection needing to transmit data. At connection setup time, the CM at a basestation contacts the MAC to perform an admission control check based on the QoS requirements of the connection. Similarly, following a handoff, the CM at a basestation contacts the MAC to verify if the QoS requirements of the newly rerouted connection can be supported.

The CM and MAC modules are designed to minimize the impact of handoffs on the performance seen by applications. We have already describe how the CM uses a smart rerouting scheme to lessen the impact of host mobility. Additionally, the MAC layer supports a mobile triggered soft-handoff technique based on the measured power from the basestation. This allows registration at a new basestation to be done before the communication with the current basestation is broken. Our protocol design also speeds up handoffs by coupling MAC-level hand-off signalling messages with CM-level hand-off signalling messages to reduce the number of messages that need to be sent over the air for handoff processing.

3.4 Error control

Our protocol architecture provides support for reliable packet transmission (ARQ) through both end to end packet retransmission and link level retransmission over the wireless last hop. Both error recovery mechanisms use selective acknowledgements and retransmission. The link level protocols also provide support for Forward Error Correction (FEC). The CM informs the LLC module at connection setup time whether to do FEC or ARQ or both on a per-VC basis. We are currently in the process of experimenting to see how critical link-level error recovery is in providing good end to end throughput. Unlike TCP, we do not marry error control to flow control and thereby avoid many of the recent reported problems with TCP throughput degradation over wireless networks [3].

3.5 API

The API provided to the SWAN protocol suite provides for a client-server style of communication. A server application at a host *binds* itself to a well known port and then *listens* for incoming connection setup requests. Once it accepts a request it can *send* or *recv* data. Similarly, the client

side issues a *connect* request to initiate the connection establishment process, after binding itself to a port. These primitives are similar in spirit to existing APIs such as Berkeley sockets.

However, our API provides several additional features. First, unlike existing interfaces, applications are allowed to specify their QoS requirements in terms of the average and peak bandwidth as well as the desired end to end latency. This information is used for doing admission control and to control the handoff/rerouting process. Second, applications are allowed to specify whether or not they require reliable data transport. Finally, applications are allowed to register their interest in being informed about specific events of interest, such as a reduction in available bandwidth, a move to a new cell etc. The CM which is the repository of state information uses a callback mechanism to notify the application when such an event occurs. The intent is to provide support for context-aware and adaptive applications [8] which can and would like to modify their behavior based on the availability of resources, current geographical area etc.

4 Status

We have a preliminary version of the network protocols, described in the paper, operational. We currently provide support for native mode ATM communication over a single hop link. In the current implementation, the CM and DT modules are implemented as a single user level process. The API is provided via UNIX IPC primitives. We have successfully ported various applications such as *nv* to run over our API. We are currently able to get about 160 Kbps of end to end user throughput and 190 Kbps of over-the-air throughput in each direction. This figure excludes the various protocol headers and represents a fairly efficient use of the current SWAN radio capacity of 625 Kbps. We are currently in the process of adding support for mobility and expect to have the complete system operational in a few months.

References

- [1] P. Agrawal et al. A Testbed for Mobile Networked Computing. In *Proceedings of 1995 IEEE International Conference on Communications (ICC '95)*, pp. 410-416, Seattle, Washington, June 1995.
- [2] P. Agrawal et al. Multimedia Information Processing in the SWAN Mobile Networked Computing System. In *Proceedings of Multimedia Computing and Networking*, San Jose, California, Jan 1996.
- [3] R. Caceres and L. Iftode. The effects of mobility on reliable transport protocols. In *International Conference on Distributed Computing System*, Cracow, Poland, June 1994.
- [4] B. Barringer, T. Burd, et. al. Infopad: A system design for portable multimedia access. In *Wireless 1994*, Calgary, Canada, July 1994.
- [5] P. P. Mishra, and M. B. Srivastava. Call Establishment and Rerouting in Mobile Computing Networks. *AT&T Bell Laboratories Technical Memorandum*, 11384-940906-13TM, September 1994.

- [6] A. Bar-Noy, I. Kessler and M. Sidi. Mobile users: To update or not to update. In *Proceedings of the Conference on Computer Communications (IEEE INFOCOM)*, pp 570-576, Toronto, Canada, June 1994.
- [7] Medium Access Control and Air-Interface Subsystem for an Indoor Wireless ATM Network, In *Proceedings of the 9th International Conference on VLSI Design*, Bangalore, India, Jan 1996.
- [8] B. Schilit, N. Adams and R. Want. Context-Aware Computing Applications. In *Workshop on Mobile Computing Systems and Applications*, Santa Clara, Dec. 1994.
- [9] J. Trotter and M. Cravatts A Wireless Adapter Architecture for Mobile Computing. In *Proceedings of 2nd USENIX Symposium on Mobile and Location Independent Computing*, pp 25-31, Ann Arbor, Michigan, April 1994.
- [10] M. Weiser. Some computer science issues in ubiquitous computing. *Communications of the ACM*, 36(7):209220, July 1993.

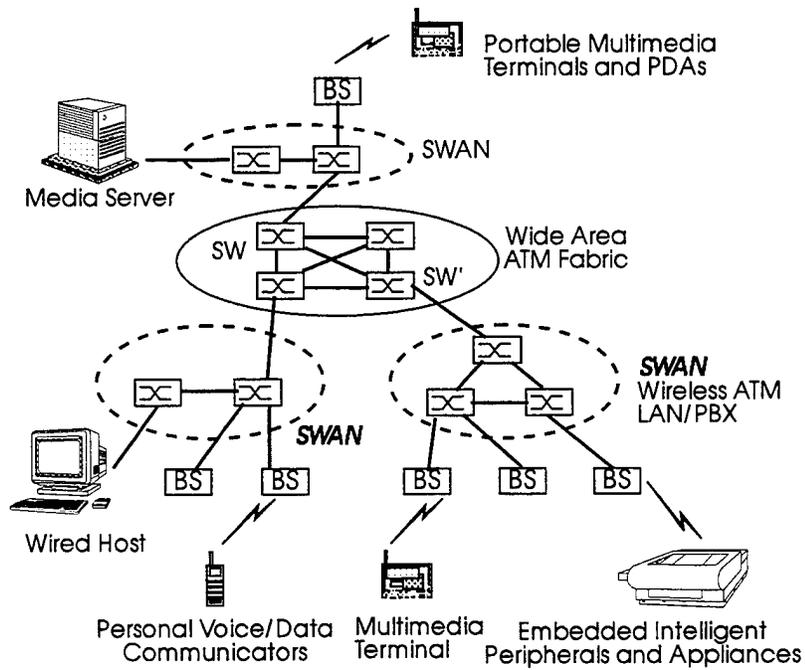


Figure 1: Network computing model

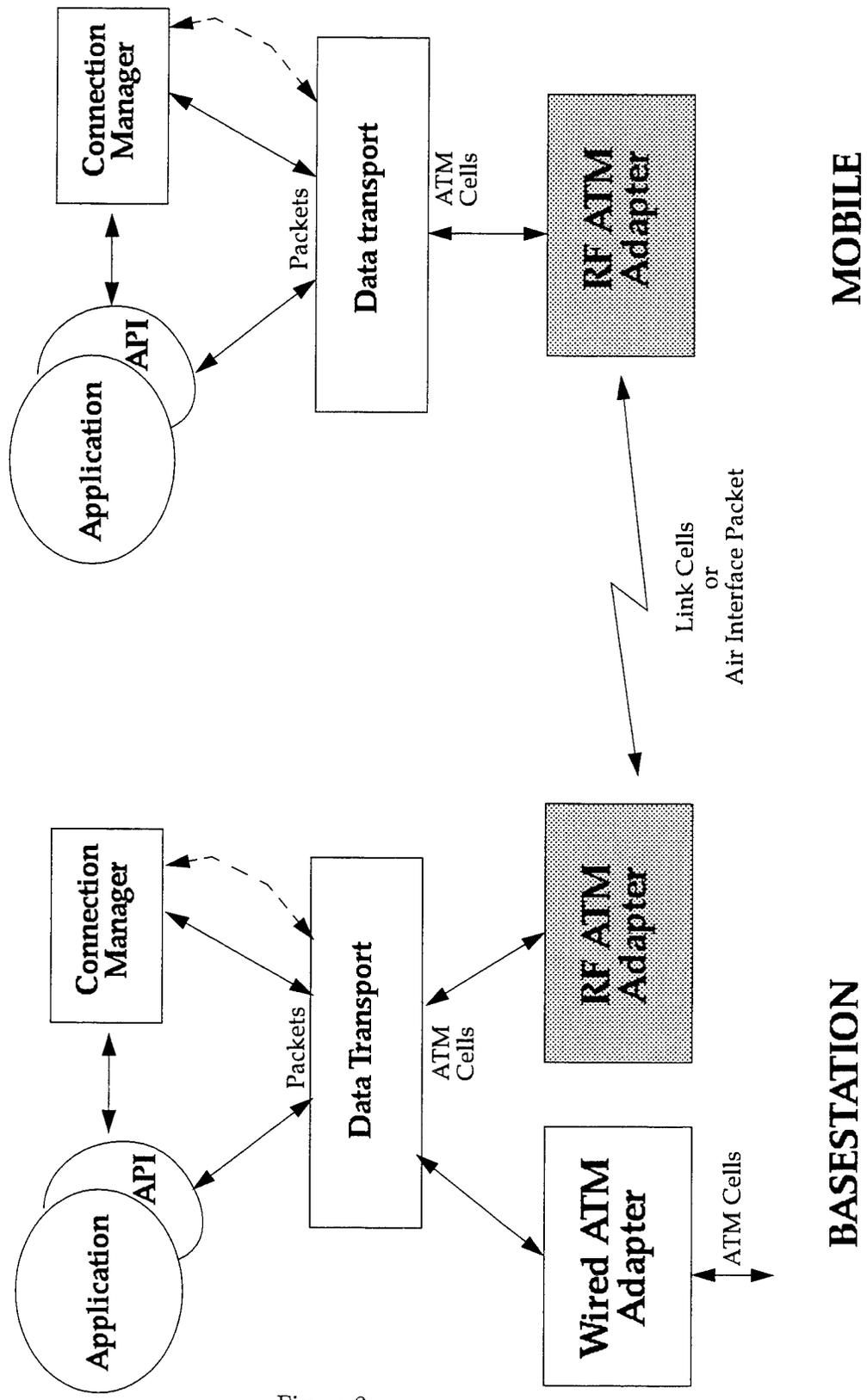


Figure 2:

Multimedia Networking¹

Jim Kurose
Department of Computer Science
University of Massachusetts
Amherst MA 01003
kurose@cs.umass.edu
<http://gaia.cs.umass.edu>

Abstract

In this talk we consider recent progress and open issues in providing networking support for real-time, interactive multimedia applications. We begin by identifying the range of network service requirements for multimedia applications such as teleconferencing, video on demand, information retrieval, and collaborative work environments. We then overview selected on-going research efforts aimed at supporting such applications in both wide area Internet- and ATM-based networks. We identify and discuss open challenges and research issues including receiver-initiated control and adaptivity, the problem of scale, the need to provide quality of service guarantees, and the attendant problems of resource reservation and call admission.

¹Copies of the overheads from this talk can be obtained from <ftp://gaia.cs.umass.edu/pub/Kuro95:Maryland.ps>

New Languages for the Integration of Heterogeneous Data Sources

Peter Buneman
University of Pennsylvania

Abstract

Much data is held not in conventional database management systems but in a variety of data formats that were designed for the express purpose for data archiving and data exchange. For example, most scientific data is held in such formats, which can vary greatly in the generality with which they can be used to represent arbitrary data. These formats cannot be easily converted to relational structures, and in some cases even object-oriented databases do not provide a natural representation. Certainly, relational query languages cannot be used in connection with these formats. Can we obtain languages for such formats that have the same nice properties possessed by relational query languages? And how can we integrate these formats with existing databases?

Ever since relational databases were first conceived, first-order logic, i.e., relational calculus/algebra has been taken as the starting point for the design of relational query languages. However, with the desire to increase the expressive power of query languages and with the need to communicate with non-relational data structures – especially those that are provided by object oriented databases and scientific data formats – I want to propose an alternative strategy: to look at the operations that are naturally associated with the data structures involved, and to use this as a guiding principle for language design. For example a database relation is a set of records. In this case, our approach is to achieve more generality and flexibility by looking independently at the canonical operations for record types and for set types. An immediate consequence of this approach is that, with the ability to combine set and record construction in an arbitrary fashion, we can build languages for “non-flat” relations, i.e. nested relations or, more generally, complex objects. Such languages are important for several reasons, among them is the ability of object-oriented databases to support objects that can themselves contain sets of objects.

A more important reason is that the same principles apply to other collection types such as lists, bags (multisets), arrays, indexed structures and certain kinds of trees. Such types are widely used in scientific data formats and in more general data exchange formats. This provides us with a general language for querying and transforming these data sources, and well-understood languages for relational databases such as relational algebra, nested relational algebra, datalog can be understood as naturally occurring fragments of this general language. An added strength of this approach is that it tells us precisely how the well-known optimizations for relational database languages can be generalized to work on other types such as lists, multisets, arrays and variants.

As an example, ASN.1 is a well-known data exchange format which – for better or worse – is used for a number of database tasks. It can be simply understood as a textual representation of data structures that can be build using the constructors for records, variants (discriminated unions), lists, and sets. It is used for example for one of the major genome databases. Using a query language based on this approach we were able to query simultaneously ASN.1 data together with information provided by relational data servers and sequence matching algorithms in order to perform certain database integration tasks that had been deemed impossible for lack of a relational representation of all the relevant data.

In this talk I shall describe this general approach and how it has been used for a variety of practical data integration tasks. If time is available I shall also describe how the same approach can be taken to "unstructured" data. Recently, formats have been proposed in which each "object" is dynamically typed through a very flexible data structure. The interpretation of the type of an object is entirely up to the user or application. However there is a very general underlying type constructor, and the same approach may be used to obtain a powerful family of languages for such structures.

This is joint work with a number of researchers in the Computer and Information Science Department and the Human Genome Center at the University of Pennsylvania.

Foundations of Multimedia Information Systems

ABSTRACT *

Sherry Marcus
21st Century Technologies, Inc.
sema@cais.com

V.S. Subrahmanian
University of Maryland
E-mail: vs@cs.umd.edu

Though numerous multimedia systems exist in the commercial market today, relatively little work has been done on developing the mathematical foundations of multimedia technology. We attempt to take some initial steps towards the development of a theoretical basis for multimedia information system. To do so, we develop the notion of a structured multimedia database system. We begin by defining a mathematical model of a media-instance. A media-instance may be thought of as “glue” residing on top of a specific physical media-representation (such as video, audio, documents, etc.) Using this “glue”, it is possible to define a general purpose logical query language to query multimedia data. This glue consists of a set of “states” (e.g. video frames, audio tracks, etc.) and “features”, together with relationships between states and/or features. A structured multimedia database system imposes a certain mathematical structure on the set of features/states. Using this notion of a structure, we are able to define indexing structures for processing queries, methods to relax queries when answers do not exist to those queries, as well as sound, complete and terminating procedures to answer such queries (and their relaxations, when appropriate). We show how a media-presentation can be generated by processing a sequence of queries, and furthermore we show when these queries are extended to include *constraints*, then these queries can not only generate presentations, but also generate temporal synchronization properties and spatial layout properties for such presentations. We describe the architecture of a prototype multimedia database system based on the principles described in this paper.

* Author for Correspondence: V.S. Subrahmanian. This research was supported by the Army Research Office under grant DAAL-03-92-G-0225, by the Air Force Office of Scientific Research under grant F49620-93-1-0065, by ARPA/Rome Labs contract Nr. F30602-93-C-0241 (Order Nr. A716), and by an NSF Young Investigator award IRI-93-57756.

RoadRunner: An Operating System for Multimedia Applications

H. Kanakia
D. Saha
S. Tripathi

Abstract

The current generation of commercially available operating systems (OSs) are rather inefficient in supporting communications and real-time data transfers. Although some progress has been made in recent years in improving the OS performance in both of these areas, we feel that a substantial shift in operating system design principles is necessary to achieve satisfactory performance. This has motivated us to propose to build an OS, named RoadRunner, that is organized around design principles different from those employed in the current generation of OS.

RoadRunner OS drops the processor-centered view common to the current generation of OSs and focuses instead on the communications between processing modules. Elevating communications between entities to a first-class status makes resource requirements for communications explicit. This allows an OS scheduler take more integrated view of computations and communications, which should lead to predictability in performance and should naturally provide performance guarantees for real-time applications. Another major difference observed is the autonomy accorded by RoadRunner OS to I/O devices. Autonomous devices, which we define as devices that can initiate I/O operations, buffer data and control data flows, communicate with each other without the intervention of a processor and its OS kernel. Thus, the OS becomes the facilitator of actions originated and terminated at autonomous devices rather than the director of I/O operations as in current systems. The shift in paradigm, characterized by communication-centered view and autonomy of devices, offers exciting opportunities and new research challenges in building OSs.

Indexing Multimedia Databases

*Christos Faloutsos**

Abstract

We describe a domain-independent method to search multimedia databases by content. Examples of these searches include ‘*find all images that look like this graphic drawing (which is a photograph of a sunset)*’ in a collection of color images; ‘*find stocks that move like Motorola’s*’ in a collection of stock price movements; and ‘*find patterns with stripes containing red and white*’ in a collection of retail catalog items.

In all these applications, we assume that there exists a distance function, which measures the dis-similarity between two objects. Given that, the idea is to extract f numerical features from each object, effectively mapping it into a point in f -dimensional space. Subsequently, any spatial access method (like the R-trees) can be used to search for similar objects (that is, nearby points in the f -d space). Comparing the features corresponds to a ‘quick and dirty’ test, which will help us exclude a large number of non-qualifying objects. The test could allow for false alarms, but no false dismissals. This implies that the mapping from objects to f -d points should preserve the distance, or, as we show, it should *lower-bound* it.

We show how this idea can be applied to achieve fast searching in color images, as well as time series. Experiments on real or realistic databases show that it is much faster than sequential scanning, while not missing any qualifying objects, as expected from the lower-bounding lemma. Thus, this approach can be used for *any* database of multimedia objects, as long as the lower-bounding lemma is satisfied.

Keywords: image databases; indexing; spatial access methods; time sequence matching.

1 Introduction

The problem we focus on is the design of fast searching methods that will search a database of multimedia objects, to locate objects that match a query object, exactly or approximately. Objects can be 2-dimensional color images, gray-scale medical images in 2-d or 3-d (eg., MRI

*With AT&T Bell Laboratories, Murray Hill, NJ; on leave from the Department of Computer Science and from the Institute for Systems Research (ISR) University of Maryland at College Park. His research was partially funded by the National Science Foundation under Grants IRI-9205273 and IRI-8958546 (PVI), with matching funds from EMPRESS Software Inc. and Thinking Machines Inc. E-address: christos@research.att.com, christos@cs.umd.edu

brain scans), 1-dimensional time series, digitized voice or music, video clips etc. A typical query by content would be, eg., '*in a collection of color photographs, find ones with a same color distribution as a sunset photograph*'.

Specific applications include image databases; financial, marketing and production time series; scientific databases with vector fields; audio and video databases, DNA/genome databases, etc. In such databases, typical queries would be '*find companies whose stock prices move similarly*', or '*find images that look like a sunset*', or '*find medical X-rays that contain something that has the texture of a tumor*'.

Searching for similar patterns in such databases as the above is essential, because it helps in predictions, computer-aided medical diagnosis and teaching, hypothesis testing and, in general, in 'data mining' [1] and rule discovery.

Of course, the distance of two objects has to be quantified. We rely on a domain expert to supply such a distance function $\mathcal{D}()$:

Definition 1 *Given two objects, O_1 and O_2 , the distance (= dis-similarity) of the two objects is denoted by*

$$\mathcal{D}(O_1, O_2) \quad (1)$$

For example, if the objects are two (equal-length) time series, the distance $\mathcal{D}()$ could be their Euclidean distance (sum of squared differences).

Similarity queries can be classified into two categories:

Whole Match: Given a collection of N objects O_1, O_2, \dots, O_N and a query object Q , we want to find those data objects that are within distance ϵ from Q . Notice that the query and the objects are of the same type: for example, if the objects are 512×512 gray-scale images, so is the query.

Sub-pattern Match: Here the query is allowed to specify only part of the object. Specifically, given N data objects (eg., images) O_1, O_2, \dots, O_N , a query (sub-)object Q and a tolerance ϵ , we want to identify the parts of the data objects that match the query. If the objects are, eg., 512×512 gray-scale images (like medical X-rays), in this case the query could be, eg., a 16×16 sub-pattern (eg., a typical X-ray of a tumor).

Additional types of queries include the '*nearest neighbors*' queries (eg., '*find the 5 most similar stocks to IBM's stock*') and the '*all pairs*' queries or '*spatial joins*' (eg., '*report all the pairs of stocks that are within distance ϵ from each other*'). Both the above types of queries can be supported by our approach: As we shall see, we reduce the problem into searching for multi-dimensional points, which will be organized in R-trees; in this case, nearest-neighbor search can be handled with a branch-and-bound algorithm (eg., [12]), and the spatial-join query can be handled with recent, highly fine-tuned algorithms [4]. Thus, we do not focus on nearest-neighbor and 'all-pairs' queries.

For all the above types of queries, the ideal method should fulfill the following requirements:

- it should be *fast*. Sequential scanning and distance calculation with each and every object will be too slow for large databases.

- it should be ‘*correct*’. In other words, it should return all the qualifying objects, without missing any (i.e., no ‘false dismissals’). Notice that ‘false alarms’ are acceptable, since they can be discarded easily through a post-processing step.
- the proposed method should require a small space overhead.
- the method should be dynamic. It should be easy to insert, delete and update objects.

As we see next, the heart of the proposed approach is to use f feature extraction functions, to map objects into points in f -dimensional space; thus, we can use highly fine-tuned database spatial access methods to accelerate the search. The remainder of the paper is organized as follows. Section 2 gives some background material on past related work, on image indexing and on spatial access methods. Section 3 describes the main ideas for the proposed, generic approach to indexing multimedia objects. Section 4 summarizes the conclusions and lists problems for future research.

2 Survey

As mentioned in the abstract, the idea is to map objects into points in f -d space, and to use multi-attribute access methods (also referred to by *Spatial Access Methods* (SAMs)) to cluster them and to search for them. There are two questions: (a) how to derive good features and (b) how to organize them in SAMs.

In terms of features to use, research in image databases benefits from the large body of work in machine vision on feature extraction and similarity measures see e.g., [2, 5]. There is also a lot of work from the database end (see, eg., [9]). In many cases, the article emphasizes either the vision aspects of the problem, or the indexing issues. Several papers (eg., [11]) comment on the need for increased communication between the vision and the database communities for such problems. See [6] for a survey of papers on feature extraction from the machine vision research, as well as on database indexing methods.

A brief introduction to multidimensional indexing methods (or Spatial Access Methods - ‘SAM’s) follows: The prevailing methods form three classes [10]: (a) R^* -trees [3] and the rest of the R-tree family [8] (b) linear quadtrees and (c) grid-files.

Several of these methods explode exponentially with the dimensionality, eventually reducing to sequential scanning. For linear quadtrees, the effort is proportional to the hypersurface of the query region [6]; the hypersurface grows exponentially with the dimensionality. Grid files face similar problems, since they require a directory that grows exponentially with the dimensionality. The R-tree based methods seem to be most robust for higher dimensions, provided that the fanout of the R-tree nodes remains > 2 .

In the subsequent work we used the R^* -tree [3] as the underlying SAM. Of course, in case that a new, faster method is invented in the future, our approach can readily take advantage of it.

3 Basic idea

To illustrate the basic idea, we shall focus on ‘whole match’ queries. There, the problem is defined as follows:

- we have a collection of N objects: O_1, O_2, \dots, O_N
- the distance/dis-similarity between two objects (O_i, O_j) is given by the function $\mathcal{D}(O_i, O_j)$, which can be implemented as a (possibly, slow) program
- the user specifies a query object Q , and a tolerance ϵ

Our goal is to find the objects in the collection that are within distance ϵ from the query object. An obvious solution is to apply sequential scanning: For each and every object O_i ($1 \leq i \leq N$), we can compute its distance from Q and report the objects with distance $\mathcal{D}(Q, O_i) \leq \epsilon$.

However, sequential scanning may be slow, for two reasons:

1. the distance computation might be expensive. For example, the editing distance in DNA strings requires a dynamic-programming algorithm, which grows like the product of the string lengths (typically, in the hundreds or thousands, for DNA databases).
2. the database size N might be huge.

Thus, we are looking for a faster alternative. The proposed approach is based on two ideas, each of which tries to avoid each of the two disadvantages of sequential scanning:

- a ‘quick-and-dirty’ test, to discard quickly the vast majority of non-qualifying objects (possibly, allowing some false-alarms)
- the use of Spatial Access Methods, to achieve faster-than-sequential searching, as suggested by Jagadish [9].

The case is best illustrated with an example. Consider a database of time series, such as yearly stock price movements, with one price per day. Assume that the distance function between two such series S and Q is the Euclidean distance

$$\mathcal{D}(S, Q) \equiv \left(\sum_{i=1} (S[i] - Q[i])^2 \right)^{1/2} \quad (2)$$

where $S[i]$ stands for the value of stock S on the i -th day. Clearly, computing the distance of two stocks will take 365 subtractions and 365 squarings in our example.

The idea behind the ‘quick-and-dirty’ test is to characterize a sequence with a single number, which will help us discard many non-qualifying sequences. Such a number could be, eg., the average stock price over the year: Clearly, if two stocks differ in their averages by a large margin, it is impossible that they will be similar. The converse is not true, which is exactly the reason we may have false alarms. Numbers that contain some information about a sequence (or a multimedia object, in general), will be referred to as ‘*features*’ for the rest of this paper. Using a good feature (like the ‘average’, in the stock-prices example), we can have a quick test, which will discard many stocks, with a single numerical comparison for each sequence (a big gain over the 365 subtractions and squarings that the original distance function requires).

If using one feature is good, using two or more features might be even better, because they may reduce the number of false alarms (at the cost of making the ‘quick-and-dirty’ test a bit more elaborate and expensive). In our stock-prices example, additional features might be, eg., the standard deviation, or, even better, some of the discrete Fourier transform (DFT) coefficients (see [7]).

The end result of using f features for each of our objects is that we can map each object into a point in f -dimensional space. We shall refer to this mapping as $\mathcal{F}()$ (for ‘F’eature):

Definition 2 Let $\mathcal{F}()$ be the mapping of objects to f -d points, that is $\mathcal{F}(O)$ will be the f -d point that corresponds to object O .

This mapping provides the key to improve on the second drawback of sequential scanning: by organizing these f -d points into a spatial access method, we can cluster them in a hierarchical structure, like the R^* -trees. Upon a query, we can exploit the R^* -tree, to prune out large portions of the database that are not promising. Such a structure will be referred to by F -index (for ‘Feature index’). Thus, we do not even *have* to do the quick-and-dirty test on all of the f -d points!

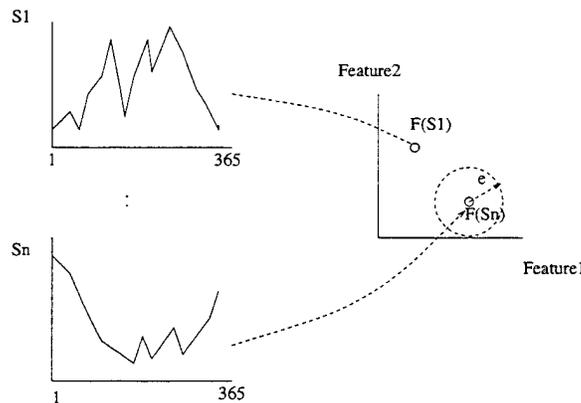


Figure 1: Illustration of basic idea: a database of sequences S_1, \dots, S_n ; each sequence is mapped to a point in feature space; a query with tolerance ϵ becomes a sphere of radius ϵ .

Figure 1 illustrates the basic idea: Objects (eg., time series that are 365-points long) are mapped into 2-d points (eg., using the average and the standard-deviation as features). Consider the ‘whole match’ query that requires all the objects that are similar to S_n within tolerance ϵ : this query becomes an f -d sphere in feature space, centered on the image $\mathcal{F}(S_n)$ of S_n . Such queries on multidimensional points is exactly what R-trees and other SAMs are designed to answer efficiently. More specifically, the search algorithm for a whole match query is as follows:

Algorithm 1 Search an F-index:

1. map the query object Q into a point $\mathcal{F}(Q)$ in feature space
2. using the SAM, retrieve all points within the desired tolerance ϵ from $\mathcal{F}(Q)$.

3. retrieve the corresponding objects, compute their actual distance from Q and discard the false alarms.

Intuitively, an F-index has the potential to relieve both problems of the sequential scan, presumably resulting into much faster searches. The only step that we have to be careful with is that the mapping $\mathcal{F}()$ from objects to f -d points does not distort the distances. Let $\mathcal{D}()$ be the distance function of two objects, and $\mathcal{D}_{feature}()$ be the (say, Euclidean) distance of the corresponding feature vectors. Ideally, the mapping should preserve the distances exactly, in which case the SAM will have neither false alarms nor false dismissals. However, requiring perfect distance preservation might be difficult: For example, it is not obvious which features we have to use to match the editing distance between two DNA strings. Even if the features are obvious, there might be practical problems: for example, in the stock-price example, we could treat every sequence as a 365-dimensional vector; although in theory a SAM can support an arbitrary number of dimensions, in practice they all suffer from the ‘dimensionality curse’, as discussed in the survey section.

The crucial observation is that we can guarantee that the ‘F-index’ method will not result in any false dismissals, if the distance in feature space matches or underestimates the distance between two objects. Intuitively, this means that our mapping $\mathcal{F}()$ from objects to points *should make things look closer* (ie., it should be a contractive mapping).

Mathematically, let O_1 and O_2 be two objects (e.g., same-length sequences) with distance function $\mathcal{D}()$ (e.g., the Euclidean distance) and $\mathcal{F}(O_1)$, $\mathcal{F}(O_2)$ be their feature vectors (e.g., their first few Fourier coefficients), with distance function $\mathcal{D}_{feature}()$ (e.g., the Euclidean distance, again). Then we have:

Lemma 1 *To guarantee no false dismissals for whole-match queries, the feature extraction function $\mathcal{F}()$ should satisfy the following formula:*

$$\mathcal{D}_{feature}(\mathcal{F}(O_1), \mathcal{F}(O_2)) \leq \mathcal{D}(O_1, O_2) \quad (3)$$

Proof: Let Q be the query object, O be a qualifying object, and ϵ be the tolerance. We want to prove that if the object O qualifies for the query, then it will be retrieved when we issue a range query on the feature space. That is, we want to prove that

$$\mathcal{D}(Q, O) \leq \epsilon \Rightarrow \mathcal{D}_{feature}(\mathcal{F}(Q), \mathcal{F}(O)) \leq \epsilon \quad (4)$$

However, this is obvious, since

$$\mathcal{D}_{feature}(\mathcal{F}(Q), \mathcal{F}(O)) \leq \mathcal{D}(Q, O) \leq \epsilon \quad (5)$$

Thus, the proof is complete. \square

We have just proved that lower-bounding the distance works correctly for range queries. Will it work for the other queries of interest, like ‘all-pairs’ and ‘nearest neighbor’ ones? The answer is affirmative in both cases: An ‘all-pairs’ query can easily be handled by a ‘spatial join’ on the points of the feature space: using a similar reasoning as before, we see that the resulting set of pairs will be a superset of the qualifying pairs. For the nearest-neighbor query, the following algorithm guarantees no false dismissals: (a) find the point $\mathcal{F}(P)$ that is the nearest neighbor to the query point $\mathcal{F}(Q)$ (b) issue a range query, with query object Q and radius $\epsilon = \mathcal{D}(Q, P)$ (ie, the actual distance between the query object Q and data object P).

In conclusion, the proposed generic approach to indexing multimedia objects for fast similarity searching is as follows (named ‘*GEMINI*’ for *GE*neric *M*ultimedia object *I*NDexIng):

Algorithm 2 (‘GEMINI’) GENEric Multimedia object INdexIng approach:

1. determine the distance function $\mathcal{D}()$ between two objects
2. find one or more numerical feature-extraction functions, to provide a ‘quick and dirty’ test
3. prove that the distance in feature space *lower-bounds* the actual distance $\mathcal{D}()$, to guarantee correctness
4. use a SAM (eg., an R^* -tree), to store and retrieve the f -d feature vectors

The first two steps of GEMINI deserve some more discussion: The first step involves a domain expert. The methodology focuses on the *speed* of search only; the quality of the results is completely relying on the distance function that the expert will provide. Thus, GEMINI will return *exactly the same* response-set (and therefore, the same quality of output, in terms of precision-recall) with what the sequential scanning of the database would provide; the only difference is that GEMINI will be faster.

The second step of GEMINI requires intuition and imagination. It starts by trying to answer the question (referred to as the ‘*feature-extracting*’ question for the rest of this work):

‘Feature-extracting’ question: *If we are allowed to use only one numerical feature to describe each data object, what should this feature be?*

The successful answers to the above question should meet two goals: (a) they should facilitate step 3 (the distance lower-bounding) and (b) they should capture most of the characteristics of the objects.

The above approach has been used successfully for retrieval by content in

- time sequences [7] and
- color images [6]

4 Conclusions

We have presented a generic method (the ‘*GEMINI*’ approach) to accelerate queries by content on image databases and, more general, on multimedia databases. Target queries are, eg., ‘*find images with a color distribution of a sunset photograph*’; or, ‘*find companies whose stock-price moves similarly to a given company’s stock*’.

The method expects a distance function $\mathcal{D}()$ (given by domain experts), which should measure the dis-similarity between two images or objects O_1, O_2 . We mainly focus on *whole match* queries (that is, *queries by example*, where the user specifies the ideal object and asks for all objects that are within distance ϵ from the ideal object). Extensions to other types of queries (nearest neighbors, ‘all pairs’ and sub-pattern match) are briefly discussed.

The ‘*GEMINI*’ approach combines two ideas:

- The first is to devise a ‘*quick and dirty*’ test, which will eliminate several non-qualifying objects. To achieve that, we should extract f numerical features from each object, which should somehow describe the object (for example, the first few DFT coefficients for a time sequence, or for a gray-scale image). The key question to ask is ‘*If we are allowed to use only one numerical feature to describe each data object, what should this feature be?*’
- The second idea is to further accelerate the search, by organizing these f -dimensional points using state-of-the art spatial access methods (‘SAMs’) [9], like the R^* -trees. These methods typically group neighboring points together, thus managing to discard large un-promising portions of the address space early.

The above two ideas achieve fast searching. We went further, and we considered the condition under which the above method will be not only fast, but also *correct*, in the sense that it will not miss any qualifying object (false alarms are acceptable, because they can be discarded, with the obvious way). Specifically, we proved the *lower-bounding* lemma, which intuitively states that the mapping $\mathcal{F}()$ of objects to f -d points should *make things look closer*.

The rest of the paper shows how to apply the method for a variety of environments, like 2-d color images and 1-d time sequences. These environments are specifically chosen, because they give rise to the ‘cross-talk’ and the ‘dimensionality-curse’ problems, respectively. The philosophy of the ‘quick-and-dirty’ filter, together with the ‘lower-bounding’ lemma, provided solutions to both cases. Experimental results on real or realistic data confirmed both the correctness as well as the speed-up that our approach provides.

Future work involves the application of the method in other, diverse environments, like voice and video databases, DNA databases, etc.. The interesting problems in these applications are to find the details of the distance functions in each case, and to design features that will lower-bound the corresponding distance tightly.

References

- [1] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. *Proc. of VLDB Conf.*, pages 487–499, September 1994.
- [2] D. Ballard and C. Brown. *Computer Vision*. Prentice Hall, 1982.
- [3] N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger. The r^* -tree: an efficient and robust access method for points and rectangles. *ACM SIGMOD*, pages 322–331, May 1990.
- [4] Thomas Brinkhoff, Hans-Peter Kriegel, Ralf Schneider, and Bernhard Seeger. Multi-step processing of spatial joins. *ACM SIGMOD*, pages 197–208, May 1994.
- [5] R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis*. Wiley, New York, 1973.

- [6] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, and W. Equitz. Efficient and effective querying by image content. *Journal of Intell. Inf. Systems*, 3(3/4):231–262, July 1994.
- [7] Christos Faloutsos, M. Ranganathan, and Yannis Manolopoulos. Fast subsequence matching in time-series databases. *Proc. ACM SIGMOD*, pages 419–429, May 1994. ‘Best Paper’ award; also available as CS-TR-3190, UMIACS-TR-93-131, ISR TR-93-86.
- [8] A. Guttman. R-trees: a dynamic index structure for spatial searching. *Proc. ACM SIGMOD*, pages 47–57, June 1984.
- [9] H.V. Jagadish. A retrieval technique for similar shapes. *Proc. ACM SIGMOD Conf.*, pages 208–217, May 1991.
- [10] Ibrahim Kamel and Christos Faloutsos. Hilbert R-tree: An Improved R-tree Using Fractals. In *Proceedings of VLDB Conference*,, pages 500–509, Santiago, Chile, September 1994.
- [11] Wayne Niblack, Ron Barber, Will Equitz, Myron Flickner, Eduardo Glasman, Dragutin Petkovic, Peter Yanker, Christos Faloutsos, and Gabriel Taubin. The qbic project: Querying images by content using color, texture and shape. *SPIE 1993 Intl. Symposium on Electronic Imaging: Science and Technology, Conf. 1908, Storage and Retrieval for Image and Video Databases*, February 1993. Also available as IBM Research Report RJ 9203 (81511), Feb. 1, 1993, Computer Science.
- [12] Nick Roussopoulos, Steve Kelley, and F. Vincent. Nearest Neighbor Queries. *Proc. of ACM-SIGMOD*, pages 71–79, May 1995.

Adaptive Query Reformulation in Content-based Image Retrieval

Gwang S. Jung	Venkat N. Gudivada
Department of Computer Science	School of EECS
Jackson State University	Ohio University
Jackson, MS 39217	Athens, OH 45701

Abstract

A content-based image retrieval (CBIR) system is required to effectively and efficiently utilize information from image repositories. Content-based retrieval is characterized by the ability of the system to retrieve relevant images based on the actual contents of an image rather than by using the keywords assigned to the images. Content-based retrieval is facilitated by a several generic query classes. Retrieval by objective, and subjective attributes constitute two such query classes and pose special problems to the query processor. In this paper, we investigate *adaptive query reformulation* as a mechanism to deal with these problems. The query reformulation algorithm is based on the functional dependency between each image attribute and the user's relevance feedback using a theoretical framework referred to as rough set theory. The importance (or weight) of each attribute and the attribute value itself is modified in the (reformulated) query based on the degree of such functional dependencies. A prototype image retrieval system based on the proposed query reformulation method has been developed. It has been developed as a generic system that can be used across different image domains. Initial results indicate that the reformulated query using the proposed method significantly improves the retrieval effectiveness.

1 Content-based Image Retrieval

Images are produced at an ever increasing rate through various sources. As an example, NASA's Earth Observing System (EOS) is projected to receive one terabyte of data per day when fully operational. A content-based image retrieval (CBIR) system is required to effectively and efficiently utilize information from these image repositories. Content-based retrieval is characterized by the ability of the system to retrieve relevant images based on the actual contents of an image rather than by using keywords assigned to the images. The notion of relevance is dynamic and is a function of both the system user's retrieval need and context.

Previous approaches to content-based retrieval have been in one of two directions [3, 6, 2, 10]. In the first direction, image contents are modeled as simple attributes. Attributes are extracted manually and are managed within the framework of conventional database management systems. Queries are specified using these attributes. Attribute-based representation of images entails a high level of image abstraction. Generally, the higher the level of abstraction, the lesser is the scope for posing ad hoc queries on the image database. Attribute-based retrieval is advocated and advanced primarily by database researchers.

The approaches in the second direction emphasize the importance of a feature extraction/object recognition subsystem as an integral part of the image retrieval system to overcome the limitations of attribute-based retrieval. This subsystem is designed to automate the feature extraction and object recognition task at the time of image insertion into the database. However, automated approaches to object recognition are computationally expensive, difficult, and tend to be domain-specific. This direction is advanced primarily by image interpretation researchers.

Recent research on CBIR recognizes the need for synergy between these two approaches. Current approaches to CBIR differ in terms of what image features are extracted, when and how they are extracted, the level of abstraction manifested, and the degree of domain-independence desired. The following generic query classes are important to facilitate CBIR [5]: Retrieval by color, texture,

sketch, shape, volume, spatial constraints, browsing, objective attributes, subjective attributes, motion, keywords, and domain concepts. An image retrieval system that features all these query classes will have a reasonable generality to deal with diverse applications. In this paper, we are concerned with retrieval by objective, and subjective attributes only.

1.1 Objective vs. Subjective Attributes

Objective attributes are those whose interpretation doesn't vary from one system user to another. For example, number of bedrooms, total floor area, are two objective attributes in the domain of architectural design. Compared to subjective attributes (discussed below), objective attributes are more precise and do not require the domain expertise either to identify or to quantify them in new image instances. *Subjective* attributes are those whose interpretation may vary significantly from one user to another. Subjective attributes are best viewed as spanning a spectrum characterized by a left hand pole (one extreme position) and a right hand pole (the other extreme position). A user's subjectivity is then associated with a specific position on this spectrum. For example, in a mug-shot image database, the subjective attribute *eyebrow shape* may assume one of the three values: *arched*, *normal*, *straight*. The value *arched* represents, say left hand pole, while the value *straight* represents the right hand pole. Retrieval by objective, and subjective attribute queries introduce special problems to the query processor for the following reasons.

1.2 Usefulness of the Query Reformulation

Since images tend to be quite distinct from each other both in structure and semantic content, structuring the content of an image to fit the (rigid) structure of a relation (relational data model) or class (object-oriented data model) is quite difficult. Often, the users are not familiar with the logical structure of the image database. Therefore, user queries tend to be subjective, imprecise, and incomplete. Because of the visual nature, images retrieved in response to the query that is identically formulated by different users can be perceived or interpreted differently by users depending on their background and characteristics. In the case of objective attributes query, rarely does a user really know what values to assign to the attributes. When there are several objective attributes, it is often difficult to determine the importance of each attribute to specify the user retrieval need. Consider the domain of hair style images. For example, though the user might have indicated high importance to the attribute "hair length" in the initial query, the retrieved images that possess a high value for the "hair length" attribute may not be considered relevant by the user. The importance of each attribute to the user may also change after several retrieved images are reviewed by the user. Hence, it is desirable that the importance of each attribute be interactively calculated by using user's relevance feedback. Relevance feedback is obtained by asking the user to simply label an image as *relevant* or *nonrelevant*.

Presence of subjective attributes in a query also poses special problems to the query processor. Consider the mug-shot image database. For a given image, the indexer might have assigned the value *normal* for the *eyebrow shape*, while from the view point of the retrieval user the value should have been *arched*. This necessitates the query processor to evaluate the query from the view point of individual users. In this paper, we propose adaptive query reformulation as a mechanism to deal with problems that arise in queries involving objective, and subjective attributes..

Under this scheme, the system first retrieves a set of images for a user's (initial) query. These images may not be relevant to the user for reasons discussed earlier. The system then obtains user relevance feedback on these images. An inductive learning module utilizes this relevance feedback to incrementally and adaptively reformulate the query to improve retrieval effectiveness. The

method incorporates techniques similar to those of vector space information retrieval for unstructured natural language text [9, 11]. The query reformulation algorithm is based on the functional dependency between each image attribute and the user's relevance feedback using a theoretical framework referred to as rough set theory [7, 8]. The importance (or weight) of each attribute and the attribute value itself in the (reformulated) query is modified based on the degree of such functional dependencies. Hence, our query reformulation algorithm is designed systematically, and the query reformulation process is both intuitive and easily understood.

We have developed a prototype image retrieval system based on the proposed query reformulation method to evaluate its effectiveness. Although the method is illustrated in hair style image domain, the method is a general one and can be used in other image retrieval domains. In fact, our prototype has been developed as a generic and extensible system that can be easily evolved by incorporating additional image domains and query reformulation algorithms. Initial results indicate that the reformulated query generated by our method significantly improves the retrieval effectiveness.

Query reformulation in image retrieval applications is investigated in [1]. However, this method needs statistics which can only be obtained by having global term occurrences in the descriptions of all the images in the database. Whenever the database is updated, such statistics must be computed. Furthermore, their query reformulation method is rather ad hoc and is difficult to interpret.

The remainder of the paper is organized as follows. In section 2, rough set theory is briefly explained. Representation of image descriptions, user query formulation tools, and the image retrieval algorithm are introduced in section 3. This section also discusses the query reformulation algorithm and the image browsing tool for eliciting user's relevance feedback. Section 4 presents preliminary experimental results and section 5 concludes the paper.

2 Rough Set Theory

Rough Set Theory (RST) proposed by Pawlak provides a systematic framework for the study of problems arising from imprecise and insufficient knowledge [7, 8].

2.1 Approximation Space

First, we describe the basic concepts of RST. Let U denote a finite set of objects, and let $R \subseteq U \times U$ be an *equivalence relation* on U . $A = (U, R)$ is called an approximation space. If $(x, y) \in R$, then x is usually related to y in a certain way and we denote this as xRy . If x and y are elements of U and (x, y) is an element of R then we can say that x and y are *indistinguishable* in the approximation space. R is often referred to as an *indiscernibility* relation.

Let $R^* = \{X_1, X_2, \dots, X_n\}$ denote the partition induced by the equivalence relation R on U , where X_i is an equivalence class (or elementary set) in R^* . For any subset $X \subseteq U$, we can define the lower and upper approximations of X , which are denoted $\underline{A}(X)$ and $\bar{A}(X)$ respectively, in the approximation space $A = (U, R)$, as follows: $\underline{A}(X) = \bigcup_{X_i \subseteq X} X_i$: the union of all elementary sets in

A that are contained in X and $\bar{A}(X) = \bigcup_{X_i \cap X \neq \emptyset} X_i$: the union of all elementary sets in A each of

which have a non-empty intersection with X . Given a subset $X \subseteq U$ representing a certain concept of interest, we can characterize the concept X in the approximations space $A = (U, R)$ with three regions: Positive Region: $POS_A(X) = \underline{A}(X)$; Boundary Region: $BND_A(X) = \bar{A}(X) - \underline{A}(X)$; Negative Region: $NEG_A(X) = U - \bar{A}(X)$. X is considered *definable* in A if $\underline{A}(X) = \bar{A}(X)$ (or $BND_A(X) = 0$); otherwise, X is said to be *non-definable* or a *rough set*. Intuitively, for example,

$POS_A(X)$ consists of several elementary concepts which are closely related to concept X . In other words, concept X can be well explained by the elementary sets in $POS_A(X)$.

2.2 Calculation of Functional Dependency Between Attributes

A simple Knowledge Representation System (KRS), denoted $S = (U, C, D, V, \rho)$, is formally defined as follows: U denotes a set of objects, C is a set of conditional attributes, D is a set of decision (or action) attributes, $\rho : U \times F \rightarrow V$ is an information function, where $F = C \cup D$, $V = \bigcup_{a \in F} V_a$, and V_a is in the domain of attribute $a \in F$. Note that the restricted function, $\rho_u : F \rightarrow V$ defined by $\rho_u(a) = \rho(u, a)$ for every $u \in U$ and $a \in F$, provides the complete information about each object u in S .

An example of a KRS is given in Table 1. Information about images in the hair style domain $U = \{I_1, I_2, I_3, I_4, I_5, I_6\}$ is characterized by means of the conditional attribute set $C = \{\text{Hair Color, Hair Length, Hair Style}\}$ and the decision (or action) attribute set $D = \{\text{Relevance}\}$. The domains of the attributes are given by: Hair Color = $\{\textit{blonde, red, brunette}\}$; Hair Length = $\{\textit{short, medium, long}\}$; Hair Style = $\{\textit{rounded, straight, feathered}\}$; Relevant = $\{\textit{yes, no}\}$.

Given the information represented in a KRS, we want to determine the functional dependency between the conditional and decision attributes. Such a functional dependency can be used for constructing procedural knowledge such as "Under what conditions can a decision take place?" Let us assume that the 6 images in Table 1 are shown to the user, and the relevance feedback/judgment with respect to each of these images given by the user is as shown in the last column of this table. In the context of example presented in Table 1, we are particularly interested in assessing the degree of functional dependency between the conditional attributes and the decision attribute (i.e., the user's relevance judgment) to know why the user makes such a decision. To answer this question, we further need to define the following terminology. For any subset G of conditional attributes C or decision attributes D , the equivalence relation on U can be defined as: $(u_i, u_j) \in \tilde{G}$ if and only if $\rho(u_i, g) = \rho(u_j, g)$ for every $g \in G$. Let $A \subseteq C$, $B \subseteq D$, and let $A^* = \{X_1, X_2, \dots, X_n\}$ and $B^* = \{Y_1, Y_2, \dots, Y_n\}$ be the partitions induced by the equivalence relation \tilde{A} and \tilde{B} , respectively. The partition B^* as a whole can be approximated or characterized by partition A^* . The quality of such an approximation depends on the relationship of the two subsets of attributes A and B . The measure of dependency of B on A is defined as [7]:

$$0 \leq \gamma_A(B) = |POS_A(B^*)|/|U| \leq 1, \quad (1)$$

where $||$ denotes the cardinality of a set, and $POS_A(B^*) = \bigcup_{Y_i \in B^*} APS(Y_i)$ in the approximation space $APS = (U, \tilde{A})$. Note that $\gamma_A(B) = 1$ when B is totally dependent of A (i.e., A functionally determines B). If $0 \leq \gamma_A(B) \leq 1$, we say that B *roughly* depends on A . A and B are *totally independent* of each other when $\gamma_A(B) = 0$. In general, the dependency of B on A can be denoted by $A \xrightarrow{\lambda} B$. For instance, from Table 1, the following can be obtained:

$$\begin{aligned} \{\text{Hair Color, Hair Length, Hair Style}\} &\xrightarrow{1.0} \{\text{Relevance}\}, \\ \{\text{Hair Color, Hair Length}\} &\xrightarrow{1.0} \{\text{Relevance}\}, \\ \{\text{Hair Color}\} &\xrightarrow{0.5} \{\text{Relevance}\}, \quad \{\text{Hair Style}\} \xrightarrow{0.33} \{\text{Relevance}\}. \end{aligned}$$

This means that the knowledge represented by the values of the condition attributes "Hair Color," "Hair Length," and "Hair Style" are sufficient to determine the relevance of an image to the user. It should also be noted that the condition attribute "Hair Style" is redundant with respect

Image	Hair Color	Hair Length	Hair Style	Relevance
I_1	<i>blonde</i>	<i>short</i>	<i>rounded</i>	<i>yes</i>
I_2	<i>brunette</i>	<i>short</i>	<i>straight</i>	<i>no</i>
I_3	<i>blonde</i>	<i>long</i>	<i>straight</i>	<i>no</i>
I_4	<i>brunette</i>	<i>long</i>	<i>feathered</i>	<i>no</i>
I_5	<i>blonde</i>	<i>medium</i>	<i>feathered</i>	<i>yes</i>
I_6	<i>red</i>	<i>medium</i>	<i>rounded</i>	<i>no</i>

Table 1: An Example of a Knowledge Representation System

	Color	Intensity	Curly	Length	Part	Cut
Characteristics	Symbolic	Numeric	Numeric	Numeric	Symbolic	Symbolic
Domain	{blonde, brunette, red}	0-100% darkness	0-100% curly	0-49 inches	{left, right, center, none}	{feathered, straight, round}

Table 2: Attributes and Their Domains in Hair Style Images

to the decision attribute "Relevant" because the removal of the latter from the KRS would not affect the dependency between the set of condition and decision attributes in this example.

3 Adaptive Image Retrieval System

An Adaptive Image Retrieval (AIR) system which incorporates query reformulation based on user relevance feedback has been developed for hair style image domain. In this section, the representation of the images and the user query, and retrieval and query reformulation algorithms of AIR system are discussed. Our discussion is limited to objective attributes.

3.1 Representation of Images and the User Query

Each database image is logically represented by n objectively measurable attributes. Then, an image can be described by an n dimensional vector as: $I_k = (a_{k1}, a_{k2}, \dots, a_{kn})$, where a_{kj} represents the value of the attribute j in image k . User query is also represented in a similar manner. However, the importance of each attribute need to be added to the user query representation. A user query q is represented as: $q^T = ((w_1, q_1), (w_2, q_2), \dots, (w_n, q_n))$, where w_j represents the importance of attribute j and q_j represents the value of attribute j . Each attribute can have either a symbolic value or a real value. For instance, a possible value for the attribute "Hair Color" can be a symbolic value such as *blonde*. A possible value for the attribute "Hair Length" can be a numeric value such as 10.5 inches. In the AIR system, each hair style image is represented by 6 attributes. Table 2 shows these attributes and their domains.

3.2 Retrieval Algorithm

Users queries are formulated using the query formulation tool. This tool is used for eliciting information from the user about the type, value, and importance of the attributes he/she would like to see in an image. The query is then translated into a vector whose components represent the values of the attributes and their importance in the query.

Algorithm RetrieveImages

```
I ← Set of images in the collection
A ← Set of attributes used to describe the images in the domain
for each  $I_k \in I$  do
begin
   $RSV_q(I_k) \leftarrow 0.0$ 
  for each  $j \in A$  do
begin
  if  $j$  is a symbolic attribute do
    if  $q_j = a_{kj}$  then  $sim(q_j, a_{kj}) = 1$  else  $sim(q_j, a_{kj}) = 0$ ;
  else if  $j$  is a numeric attribute do
begin
   $max\_range \leftarrow$  maximum value of the attribute  $j$ 
   $sim(q_j, a_{kj}) \leftarrow 1 - \frac{|q_j - a_{kj}|}{max\_range}$ 
end
   $RSV_q(I_k) \leftarrow RSV_q(I_k) + w_j \times sim(q_j, a_{kj})$ 
end
end
end RetrieveImages
```

Figure 1: Retrieval Algorithm

The retrieval function for calculating the retrieval status value (RSV) or similarity of an image I_k with respect to a user query q is defined as follows:

$$RSV_q(I_k) = \sum_{i=1}^n w_j \times sim(q_j, a_{kj}),$$

where n represents the number of attributes and w_j represents the importance of the attribute j in the query. $sim(q_j, a_{kj})$ calculates the similarity between the value of attribute j in the user query vector and the value of j in the image representation vector. The retrieval algorithm is shown in Figure 1.

The RSV of each image with respect to the user query is then computed. The images are sorted in descending order of their RSVs. The names of images are retrieved and shown to the user in this order. Image browsing tool is used to view the retrieved images (Figure 2). This tool is also used for eliciting user's relevance feedback.

3.3 Query Reformulation Algorithm and Learning from User Feedback

As the user views images and provides relevance feedback to the system, the system creates a table internally as depicted in Table 3. Since symbolic (attribute) values are used to categorize images into equivalence classes, we need to define rules for converting real-valued attributes into symbolic values. These conversion rules are shown in Table 4. Attribute weights in the user query are recomputed (hence, query is reformulated) using RST and Table 3. A new set of images is then determined by the system using the reformulated query. The query reformulation algorithm is shown in Figure 3. $FD(j, D)$ represents the functional dependency between the attribute j and decision attribute D (i.e., Relevance). This functional dependency is calculated based on the assumption that the conditional attributes are independent of each other. Since the calculation of $POS_C(D^*)$ involves set manipulation, the computational complexity of this algorithm may be

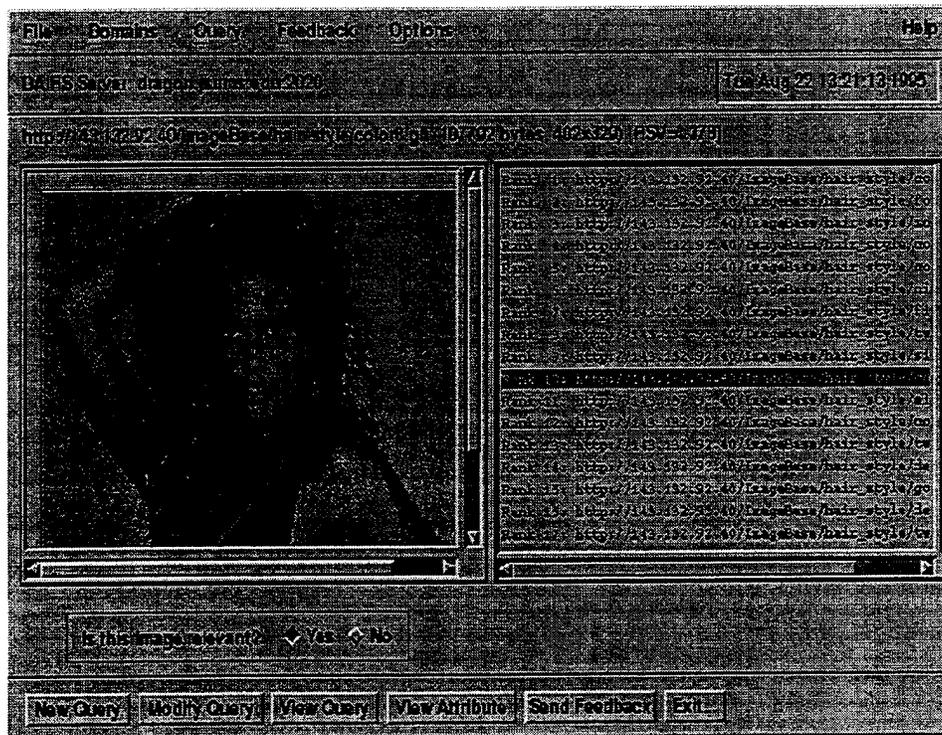


Figure 2: Image Browsing Tool of the AIR System

Image	Color	Intensity	Curly	Length	Part	Cut	Relevance
I_{rank1}	<i>blonde</i>	<i>very-dark</i>	<i>none</i>	<i>short</i>	<i>left</i>	<i>round</i>	<i>yes</i>
I_{rank2}	<i>brunette</i>	<i>light</i>	<i>light-curly</i>	<i>long</i>	<i>right</i>	<i>round</i>	<i>no</i>
I_{rank3}	<i>red</i>	<i>dark</i>	<i>curly</i>	<i>long</i>	<i>center</i>	<i>straight</i>	<i>yes</i>

Table 3: Internal Table Created After the User has Viewed Three Top Ranked Images

high. However, in reality, since the number of images reviewed by the user is relatively small for the query reformulation, the algorithm runs fast enough for interactive use. The user involvement in query reformulation is at relatively high level of abstraction and the task is quite simple. Rough Set Theory provides the formal foundation for the query reformulation, consequently, the manner in which the query reformulation is performed is systematic and easily justified.

4 Preliminary Experimental Results And Discussion

Two experiments were conducted to test the effectiveness of query reformulation in the AIR system. These trials were set up to show the ability of the system to modify the initial query to the point where attribute weights have reversed themselves due to feedback. The experiment was conducted with an image collection consisting of 80 images in the hair style domain. In both test runs, one specific attribute was chosen and the user assigned weight for that attribute was set to zero. The user was then asked to provide relevance feedback based solely on the closeness of the match between the query and the image with regard to the attribute that was chosen. The system should “recognize” the fact that the user is selecting images as relevant based solely upon the attribute whose weight was assigned initially to zero. The system should then “modify” the weight of this

Intensity	Curly	Length
<i>very-light</i> ($\leq 25\%$)	<i>not-curly</i> ($\leq 25\%$)	<i>short</i> (≤ 3 inches)
<i>light</i> ($\leq 50\%$)	<i>light-curly</i> ($\leq 50\%$)	<i>medium</i> (≤ 6 inches)
<i>dark</i> ($\leq 75\%$)	<i>medium-curly</i> ($\leq 75\%$)	<i>long</i> (≤ 9 inches)
<i>very-dark</i> ($\leq 100\%$)	<i>heavy-curly</i> ($\leq 100\%$)	<i>very-long</i> (≤ 49 inches)

Table 4: Rule for Converting Values of Real-Valued Attributes into Symbolic Values

Algorithm QueryReformulaton

```

A ← Set of attributes used to describe the images in the domain
n ← Number of images for which the user has provided relevance feedback
D ← {Relevance}
REL ← Set of images considered as relevant by the user
NREL ← Set of images considered as non-relevant by the user
for each j ∈ A do
  begin
    C ← {j}
    POSC(D*) ← ∅
    Obtain elementary sets C*
    for each elementary set e ∈ C* do
      if ((e ⊆ REL) or (e ⊆ NREL)) then
        POSC(D*) ← POSC(D*) ∪ e
      FD(j, D) ← |POSC(D*)|/n
      wj ← FD(j, D)
    end
  end
end QueryReformulation

```

Figure 3: Query Reformulation Algorithm

Attribute	Color	Intensity	Curly	Length	Part	Cut
Value	<i>blonde</i>	30.0	10.0	16.0	<i>right</i>	<i>feathered</i>
Weight	1.00	1.00	1.00	0.0	1.00	1.00

Table 5: User's Initial Query for the Experiment: Case 1

	I_{rank1}	I_{rank2}	I_{rank3}	I_{rank4}
Color	<i>blonde</i>	<i>blonde</i>	<i>blonde</i>	<i>blonde</i>
Intensity	30.0	40.0	43.0	10.0
Curl	5.0	8.0	2.0	12.0
Length	7.0	14.0	7.0	14.0
Part	<i>left</i>	<i>none</i>	<i>right</i>	<i>right</i>
Cut	<i>feathered</i>	<i>feathered</i>	<i>round</i>	<i>straight</i>
Relevant	<i>no</i>	<i>yes</i>	<i>no</i>	<i>yes</i>

Table 6: User's Relevance Feedback for Case 1

attribute to a greater value while decreasing the weights of the other attributes. In both test cases, the system automatically modified the weights of the selected attributes as expected.

In the first experiment (Case 1), the initial user query was formulated as shown in Table 5. The user assigned weights to attributes are shown in the third row of the table. Notice that the user has assigned a weight of 0.0 for the attribute "Length," meaning that the user initially did not care about the hair length in the images to be retrieved. Table 6 shows the top 4 images that were retrieved with respect to the initial query. The relevance feedback was provided by the user based solely on the attribute Length. Table 7 shows that while initially the importance of attribute Length was zero, after user relevance feedback only on four images, this value has been increased to 1.0. Also, notice in Table 6 that the user preferred to have images which contain the value "long" for the attribute Length. Such user preference is well reflected in the weights of the reformulated query as shown in Table 7. Table 8 shows the retrieved images with respect to the reformulated query. As is evident from this table, images which contain the value "long" for the attribute Length are retrieved. This indicates that the system is able to capture user's conceptual preference of certain images over the others.

The second experiment (Case 2) was conducted with the attribute "Curly" as a focus of attention. The weight of the attribute "Curly" has been set to zero to signify that the attribute "Curly" is unimportant to the user. The user preferences are based solely on the degree of curl. The images which have heavy curly hair style were selected relevant. The query is reformulated based on these preferences and the top 4 images retrieved are shown in Table 9. Observe that the images which have heavy curly hair style are placed at the top of the rank ordering.

Both the experiments were conducted with only a single attribute as a focus of attention. More experiments need to be conducted with multiple attributes as focus of attention in order to see the retrieval effectiveness of reformulated query in a more general case.

	Color	Intensity	Curly	Length	Part	Cut
Reformulated Query Weight	0.00	0.25	0.00	1.00	0.50	0.50

Table 7: Attributes Weights After Query Reformulation

	I_{rank1}	I_{rank2}	I_{rank3}	I_{rank4}
Color	<i>brunette</i>	<i>red</i>	<i>brunette</i>	<i>brunette</i>
Intensity	72.0	95.0	79.0	10.0
Curl	40.0	75.0	72.0	45.0
Length	16.0	16.0	16.0	16.0
Part	<i>none</i>	<i>none</i>	<i>none</i>	<i>none</i>
Cut	<i>feathered</i>	<i>feathered</i>	<i>round</i>	<i>round</i>

Table 8: Images Retrieved by the Reformulated Query-Case 1

	I_{rank1}	I_{rank2}	I_{rank3}	I_{rank4}
Color	<i>red</i>	<i>blonde</i>	<i>brunette</i>	<i>red</i>
Intensity	92.0	86.0	100.0	95.0
Curl	99.5	100.0	98.9	100.0
Length	3.0	13.8	5.0	2.0
Part	<i>center</i>	<i>center</i>	<i>none</i>	<i>none</i>
Cut	<i>straight</i>	<i>straight</i>	<i>round</i>	<i>round</i>

Table 9: Images Retrieved by the Reformulated Query-Case 2

5 Conclusions

In this paper, we have described the design and implementation of an adaptive image retrieval system that incorporates query reformulation mechanism. This mechanism allows imprecise and incomplete user query specifications. Our method incorporates techniques similar to those of vector space information retrieval for an unstructured natural language texts. The advantages of our approach to IR is that the image and query descriptions need not be highly structured as would be required with the IR based on the traditional database management techniques. Furthermore, an inductive learning module is designed for providing facilities by which a user's relevance feedback is effectively utilized to adaptively improve retrieval effectiveness.

The user involvement in query reformulation is at relatively high level of abstraction. Rough set theory provides the formal foundation for the query reformulation; consequently, the manner in which the query reformulation is performed is systematic and easily justified.

Preliminary experimental results indicate that the reformulated query generated by our method significantly improves the retrieval effectiveness. The initial results warrant further study of the retrieval effectiveness of the proposed approach. We are currently investigating how attribute values can be changed using the user relevance feedback. Also, we are examining how boundary region of the rough sets induced by the user's relevance feedback can be utilized. We are exploring probabilistic rough set theory for adaptive query reformulation especially when the user's feedback is random or chaotic. Finally, we are currently studying the effectiveness and efficiency of the proposed method more rigorously across different image domains distributed in the network [4].

Acknowledgment

This work has been supported in part by ARPA Grant Number: N00174-93-RC-00004 (first author) and in part by Ohio University Research Office under Grant Number: 917 (second author).

References

- [1] S. Al-Hawamdeh et al. Nearest neighbor searching in a picture archive system. In *International Conference on Multimedia Information Systems*, pages 19–34, McGraw-Hill, 1991.
- [2] A.D. Narasimhalu (Guest Editor). Special issue on content-based retrieval. *ACM Multimedia Systems*, 3(1), 1995.
- [3] W. Grosky and R. Mehrotra (Guest Editors). Image database management. *IEEE Computer*, 22(12), 1989.
- [4] G. Jung and V. Gudivada Distributed adaptive attribute-based image retrieval (*to appear*) *SPIE Photonics East '95 Symposium on Digital Image Storage and Archiving Systems*, Philadelphia, PA, October 1995.
- [5] V. Gudivada and V. Raghavan (Guest Editors). Content-based image retrieval systems. *IEEE Computer*, 28(9):2–6, 1995. Special Issue on Content-Based Image Retrieval Systems.
- [6] R. Jain. NSF workshop on visual information management systems. *SIGMOD Record*, 22(3):57–75, 1993.
- [7] Z. Pawlak. Rough sets. *International Journal of Information and Computer Sciences*, 11(5):145–172, 1982.
- [8] Z. Pawlak and S.K.M. Wong. Rough sets: Probabilistic versus deterministic approach. Technical report, University of Regina, Department of Computer Science, Regina, Saskatchewan, Canada, 1987.
- [9] V. Raghavan and S.K.M. Wong. A critical analysis of vector space model for information retrieval. *Journal of the American Society for Information Science*, 37(5):279–287, 1986.
- [10] A. Ralescu and R. Jain (Guest Editors). Special issue on advances in visual information management systems. *Journal of Intelligent Information Systems*, 3(3), 1994.
- [11] G. Salton. *Automatic Text Processing*. Addison-Wesley, Reading, MA, 1989.

Research Into Multimedia Database Systems

Jeffrey Ullman
Stanford University
ullman@DB.Stanford.EDU

Abstract

There is no doubt that the use of multimedia information is growing rapidly. The first question to ask is to what extent database research and/or expertise should be expected to provide solutions to the problems related to multimedia information. We identify several areas where we can be optimistic that database technology will be important: Management of tertiary storage, Development of type systems for special kinds of information, Query systems for multimedia information (including user interfaces), and support for varying qualities of service. In each of these areas we shall consider the prospects for research achievements and breakthroughs.

Query Refinement in Constraint Multimedia Databases

Extended Abstract

ELISA BERTINO AND BARBARA CATANIA

Dipartimento di Scienze dell'Informazione
Università degli Studi di Milano
Via Comelico 39/41
20135 Milano, Italy
e-mail: bertino@hermes.mc.dsi.unimi.it
e-mail: catania@ghost.dsi.unimi.it

1 Introduction

The need of using database systems in multimedia applications is rapidly growing. An important characteristic of those applications is the presence of digital images and spatial objects. Recent technological advances have resulted in a variety of tools supporting creation, scanning, transmission, storage and manipulation of digital images. Moreover, images can be easily combined using multimedia editors with other data types, such as text and voice, to generate multimedia objects.

Despite those advances, an issue still open is represented by image retrieval from large image databases. A first problem is represented by the intrinsic difficulty of the image recognition process. Recognizing semantic objects within an image is in general possible only when the application domain is very narrow and the objects that can occur in images are known in advance. Even when image objects have been identified, a second problem is that images may be retrieved according to a large variety of predicates, including predicates on image color, texture, shape, layout. Often, sketches are used to specify how the desired images look like.

The evaluation of a query predicate in an image database is therefore considerable more complex than in a traditional database, storing only formatted data. Evaluating a predicate against an image database may require for example the use of sophisticated computational geometry algorithms [17]. Recent optimization strategies defined for spatial and image databases thus rely on the notion of *query filters*. The basic idea of those filters is to keep approximate representations of images, in addition to exact representations. Those approximate representations are used by the query processor when evaluating a given predicate to quickly discard non qualifying images. The images passing the filtering phase are then evaluated again with respect to the same predicate; this evaluation is performed on the exact image representations. Note that some of the images that have passed the filtering phase may be discarded during the latter phase. Indeed, the use of approximate representations may result in *false drops*, because the set of images retrieved by the filtering phase is often a superset of the images that actually verify the queries.

In this paper, we discuss approximation strategies in the framework of constraint databases.

Constraint data models [5, 7, 13, 14] have been initially introduced as a mechanism to model infinite relations. Lately, those models have been applied to the domain of spatial data [16]. Indeed, 2- and 3-dimensional objects, typically found in spatial data and queries, can be simply represented as constraints in a constraint database. Because image objects are often approximated in terms of geometric descriptions, the use of a constraint data model to image retrieval in multimedia database appears a promising direction, which has however several open questions.

The remainder of this paper is organized as follows. Section 2 presents a constraint relational model by first introducing the notions of constraint tuple and constraint relation, and then the associated relational algebra. Section 3 discusses the use of such model in approximation strategies for image retrieval in multimedia databases. Section 4 outlines some conclusions and future work.

2 Constraint relational model

2.1 The data model

The model we consider is an extension of the model proposed in [13] and of [16]. The considered extension can be seen as a simplification from a data representation point of view.

The model is introduced by the following definition.

Definition 1 *Let U be a countably infinite domain of atomic values. Let Φ be a class of constraints (a decidable logical theory).*

- *A constraint k -tuple over variables x_1, \dots, x_k in the logical theory Φ is a finite disjunction of conjunctions. Each conjunction has the form $\varphi_1 \wedge \dots \wedge \varphi_N$, where each φ_i , $1 \leq i \leq N$, is a constraint in Φ . The variables in each φ_i are all free and among x_1, \dots, x_k .*
- *An intensional tuple of type $[n, m]$ is a tuple $(a_1, \dots, a_n; \varphi)$, where $a_1, \dots, a_n \in U$ and φ is a constraint tuple over variables x_1, \dots, x_m .*
- *A constraint relation of type $[n, m]$ in Φ is a finite set $r = \{\psi_1, \dots, \psi_M\}$ where each ψ_i , $1 \leq i \leq M$ is an intensional k -tuple of type $[n, m]$ on Φ .*
- *A constraint database is a finite set of constraint relations.*

The schema of a constraint relation R of type $[n, m]$ is the union of the n attribute names with the m variables. We denote with $\alpha(R)$ the schema of a relation R . We assume that the attribute names and the variables of a relation are two disjoint sets.

The existence of a relational part and of a constraint part inside each tuple [16] allows to bind descriptive knowledge, for example the name of an object, to an object represented by the constraint.

Example 1 *Suppose the database consists of a set of rectangles in the Euclidean plane. A possible representation in the relational model consists of a relation R , containing a tuple of the form (a, b, c, d) for each rectangle. Such tuple represents the rectangle with corners (a, b) , (a, d) , (c, b) and (c, d) . In the generalized relational model, rectangles can be represented by generalized tuples of the form $(a \leq x \leq c) \wedge (b \leq y \leq d)$. \square*

In general, we deal with constraints in *normal form* [4, 5]. A *normal form* for a constraint C is another constraint, equivalent to C , but avoiding redundancies which may be present in C . We

suppose that a generalized relation can contain several equivalent (i.e, with the same extension) generalized tuples only if their representation in normal form is different.

Each constraint tuple represents a set of relational tuples. For a given constraint tuple t , we denote with $c_ext(t)$ this set. Thus, each intensional tuple of type $[n, m]$ is a finite representation of a possibly infinite subset of $U^n \times R^m$, where R is the domain of the constraint theory on which the constraint tuple has been defined. Given an intensional tuple $(a_1, \dots, a_n; \varphi)$, the defined subset, denoted by $ext((a_1, \dots, a_n; \varphi))$, is given by $\{(a_1, \dots, a_n)\} \times c_ext(\varphi)$, where \times identifies the cartesian product. The elements of this product are called *extensional tuples*.

Remark 1 *Note that each value v for an attribute A can be seen as the constraint $A = v$. This may not be a constraint of the logical theory considered by the constraint relation. However, in order to simplify the following discussion, we assume to convert each intensional tuple in a constraint tuple. The theory of the obtained constraint tuple will be many-sorted. In particular, it has a sort for the constraint tuples and a sort for any attribute domain different from the constraint tuple theory.*

Thus, in the following we only speak of variables and constraint tuples.

2.2 The constraint relational algebra $GRA(\Phi, \mathcal{F})$

In [2] we have proposed a relational algebra for constraint databases. The algebra can be seen as an extension of the algebra defined in [16]. The proposed extensions tailor the algebra to a particular application domain. This tailoring is possible by specifying the considered logical theory and a set of functions \mathcal{F} (assumed to be recursive and total) on generalized tuples. For example, if we deal with spatial applications and, for efficiency requirements, we choose the theory of linear polynomial constraints [14], we can introduce the concept of distance by defining an ad hoc function. Note that the definition of the functions in \mathcal{F} may rely on efficient algorithms defined within a specific application domain. Formally, \mathcal{F} is a set of total recursive functions such that $\forall f \in \mathcal{F}, f : A \rightarrow B, A, B \subseteq DOM_{gentuple}(\Phi)$ and $DOM_{gentuple}(\Phi)$ is the set of all the possible generalized tuples defined on the logical constraint theory Φ (that, for remark 1, can be multi-sorted).

Each language defined in this way consists of a fixed set of operators, defined regardless of the application domain. However, the language includes an additional set of operators, strictly dependent on the chosen domain. Thus, if the generalized relational model is to be used in a spatial context, some operators strictly related to spatial queries are introduced in the language. Note that if we want to model temporal applications, the set of operators to consider may be different.

Our approach thus avoids introducing a "complex" logical theory, with high computational complexity. Moreover, it allows adopting a "simple" logic, for example the linear polynomial inequality constraint theory, and adding the specific functionalities requiring a higher complexity as functions. We denote with $GRA(\Phi, \mathcal{F})$, often abbreviated as GRA , the family of languages constructed in this way.

The second extension is related to the introduction of a new class of operators. Indeed, operators can also be distinguished by how they consider constraint tuples. Indeed, constraint tuples can be considered under two different points of view: as a finite representation of an infinite set of tuples, or as a single object, having a meaning by itself.

For example, consider a generalized relation $R(X, Y)$ where each generalized tuple represents a rectangle. Each tuple will have the form: $X \geq a_1 \wedge X \leq a_2 \wedge Y \geq b_1 \wedge Y \leq b_2$. If we want to determine the set of points contained in the intersection space of two given rectangles, each constraint should be considered as the finite representation of an infinite set of points. However, if we want to know which rectangles are contained in a given space, each constraint must be

considered as a single object. In the latter case, all the points of a single rectangle must satisfy a certain condition.

By taking these two different points of view into account, we define two sets of operators, *tuple operators* and *set operators*. Set operators consider generalized tuples as single objects and assume a sort of universal quantification over all the relational tuples representing a single constraint tuple.

The operators of our algebra are described in Table 1. The table classifies application independent operators into *tuple* and *set* operators. In the table we have also inserted some useful derived operators. Following the approach proposed in [11], each operator is described by using two kinds of clauses: those presenting the *scheme restrictions* required by the argument relation schemes and the scheme of the result relation, and those introducing the operator semantics. In the table, *attr* is a function that takes an expression and returns the set of attributes (variables) appearing in it.

Among the application dependent operators, *set selection* is certainly the most important operator. This operator selects all the generalized tuples satisfying a certain condition from a generalized relation. The condition has the form (P, θ) , where P is a generalized tuple on the considered logical theory and $\theta \in \{\subseteq, \supseteq, (\bowtie \neq \emptyset), (\bowtie = \emptyset)\}$.

The meaning is that we select only the generalized tuples such that there exists a relation θ between the extension of the generalized tuple and the extension of P . The meaning of θ operators is:

- $\theta = \subseteq$: we select all the tuples whose extension is contained in the extension of P (i.e., those such that, if $\alpha(t) = \tilde{x}$, $\forall \tilde{x} t \rightarrow P$ holds);
- $\theta = \supseteq$: we select all the tuples whose extension contains the extension of P (i.e., those such that, if $\alpha(t) = \tilde{x}$, $\forall \tilde{x} P \rightarrow t$ holds);
- $\theta = (\bowtie = \emptyset)$ [$\theta = (\bowtie \neq \emptyset)$]: we select all the tuples whose natural join with P is [is not] empty (i.e., those such that $t \wedge P$ is inconsistent [$t \wedge P$ is not inconsistent]).

Let D be a generalized database scheme. Let Φ be a decidable logical theory. Let \mathcal{F} be a set of total recursive functions. The $\text{GRA}(\Phi, \mathcal{F})$ language for D is composed of all the expressions definable by using the operators introduced in Table 1, applied to all the relations in D as well as to relations implicitly defined by constraints. Functions contained in \mathcal{F} can also be applied to constraints explicitly used in GRA expressions. For example, the expression $R(X, Y) \wedge f(Z = X + Y)$ is a $\text{GRA}(\Phi, \mathcal{F})$ expression for the database scheme containing a generalized relation R over X and Y and for $f \in \mathcal{F}$.

3 Image processing in constraint databases

A multimedia database systems deals with the storage, manipulation and retrieval of all types of digitally representable information objects, such as text, images, video and sound. An important issue in this context is the definition of efficient mechanisms that allow the user to retrieve desired multimedia information.

Because often content equality is not well-specified, special techniques are needed for the retrieval of multimedia objects with content similar to that specified in the user's query. For example in image databases, one can retrieve an image's if the image's features, such as shape and spatial positions, are close to the ones specified in the query.

In order to obtain good performance when executing both exact - i.e. equality-based queries- and similarity-based queries, a good heuristic is that of processing the query in order to transform

Operator name	Operator symbol	Scheme restrictions	Semantics
<i>Application independent operators</i>			
Tuple operators			
natural join	$r = r_1 \bowtie r_2$	$\alpha(r) = \alpha(r_1) \cup \alpha(r_2)$	$r = \{t : \exists t_1 \in r_1, \exists t_2 \in r_2, t = t_1 \wedge t_2\}$
renaming	$r = \rho_{A B}(r_1)$	$A \in \alpha(r), B \notin \alpha(r),$ $\alpha(r) = (\alpha(r_1) \setminus \{A\}) \cup \{B\}$	$r = \{t : \exists t' \in r_1, t[B] = t'[A],$ $t[C] = t'[C], C \neq B\}$
tuple difference	$r = r_1 \setminus^t r_2$	$\alpha(r_1) = \alpha(r_2) = \alpha(r)$	$r = \{t : \exists t_1 \in r_1, t = t_1 \wedge \neg t_2^1 \wedge \dots \wedge \neg t_2^n\}$ $r_2 = \{t_2^1, \dots, t_2^n\}$
tuple selection	$r = \sigma_P^t(r_1)$	$attr(P) \subseteq \alpha(r), \alpha(r) = \alpha(r_1)$	$r = \{t : \exists t_1 \in r_1, t = t_1 \wedge P\}$
Set operators			
set difference	$r = r_1 \setminus^s r_2$	$\alpha(r_1) = \alpha(r_2) = \alpha(r)$	$r = \{t : t \in r_1, t \notin r_2\}$
set selection	$r = \sigma_{(P,\theta)}^s(r_1)$	$attr(P) \subseteq \alpha(r), \alpha(r) = \alpha(r_1)$	$r = \{t : \exists t \in r, ext(t)\theta ext(P) \text{ is true}\}$
union	$r = r_1 \cup r_2$	$\alpha(r_1) = \alpha(r_2) = \alpha(r)$	$r = \{t : t \in r_1 \text{ or } t \in r_2\}$
projection	$r = \Pi_{i_1, \dots, i_p}(r_1)$	$\alpha(r_1) = \{x_1, \dots, x_m\},$ $1 \leq i_1, \dots, i_p \leq m,$ $\alpha(\Pi_{i_1, \dots, i_p}(r_1)) = \{x_{i_1}, \dots, x_{i_p}\}$	$r = \{\pi_{i_1, \dots, i_p}(t) : t \in r_1\}$ y_1, \dots, y_p are variables different from each x_i $\pi_{i_1, \dots, i_p}(t) = \exists x_1, \dots, \exists x_m (t \wedge \bigwedge_{l=1}^p y_l = x_{i_l})$
<i>Application dependent operators</i>			
transformation	$r = AT_f(r_1)$		$r = \{t : \exists t' \in r_1, f \in \mathcal{F} : t = f(t')\}$
set selection	$r = \sigma_{(f_1, f_2, \theta)}(r_1)$	$\alpha(r) = \alpha(r_1)$	$r = \{t : \exists t \in r_1, ext(f_1(t))\theta ext(f_2(t)) \text{ is true}\}$
<i>Derived operators</i>			
cartesian product	$r = r_1 \times r_2$	$\alpha(r) = \alpha(r_1) \cup \alpha(r_2), \alpha(r_1) \cap \alpha(r_2) = \emptyset$	$r = r_1 \bowtie r_2$
derived selection	$r = \sigma_{(=\emptyset)}^s(r_1)$	$\alpha(r) = \alpha(r_1)$	$r = r \cap^s inc$
	$r = \sigma_{(\neq\emptyset)}^s(r_1)$	$\alpha(r) = \alpha(r_1)$	$r = r \setminus^s \sigma_{(=\emptyset)}^s(r)$
tuple intersection	$r = r_1 \cap^t r_2$	$\alpha(r) = \alpha(r_1) = \alpha(r_2)$	$r = r_1 \bowtie r_2$
set intersection	$r = r_1 \cap^s r_2$	$\alpha(r) = \alpha(r_1) = \alpha(r_2)$	$r = r_1 \setminus^s (r_1 \setminus^s r_2)$

Table 1: Constraint algebra operators

it into a query which is as precise as possible, leading to the retrieval of only a small subset of objects for a direct content analysis.

The general approach by which the query results are generated by approximating several intermediate results is called *approximation-based query processing*. Such an approach is feasible if some approximated data structures are used to store data. Thus, two approximation levels can be devised:

1. *Structure-dependent approximation*. Data objects are approximated in some ways, with respect to their data type [6]. Queries, when posed against the approximated structures, return an approximated result that must be refined in some following steps. If an object is not generated as the result of a query when it is executed on an approximated level, the object is certainly not part of the proper query answers. In the other case (i.e. the object satisfies the query on the approximated representation), we are not sure that the object is an answer for the query. The query answers are then generated by executing again the query on a further more precise approximation of the retrieved objects.
2. *Execution-dependent approximation*. In this case, the query execution itself is approximated by solving it in different steps, each of which considers for the execution only the set of objects retrieved in the previous steps. In general, for spatial queries a two steps (execution-dependent) approximated computation is proposed (see for example [6, 15]): in the *filter step*, a superset of the response set is identified by using (structure-dependent) approximations, such as bounding box, as a geometric key. Several approximations can be used in sequence in the filter step. The *refinement step* inspects the exact representation of each object in the

superset, in order to obtain the exact query answers.

An interesting aspect is how constraint databases can be used to model image processing in a multimedia context and which are the advantages of using such an approach.

In the following we present several issues related to this topic, pointing out several aspects that should be further investigated in order for constraint databases to be practically used in a multimedia context.

Given an image, we assume the existence of a procedure able to transform the image in a constraint, spatially representing the (boundary of) the image.

The image-based dictionary is represented by a constraint relation R , containing an attribute Id , corresponding to an object identifier, two variables X and Y , representing the image in the two-dimensional space and an attribute D , containing the textual description of the image. The intensional tuples contained in R represent all the images the system is assumed to know.

In order to model sub-images relationships, we assume to have a relation S , containing two attributes Id and Sub_Id . Each tuple of this relation contains information of a certain object and one of its sub-objects.

By using this image representation, in the following we discuss how approximation-based query processing and similarity-based queries can be modeled by constraint databases.

3.1 Similarity-based queries

In order to model similarity-based queries, a notion of distance, modeling the degree of similarity between two objects is needed. For this purpose, let *Distance* be a function that takes two images and computes their distance. The distance between two images is a value representing the approximation degree of the second image with respect to the first one. For example, in [6], the considered distance is the false area of the approximation which may be either positive or negative with respect to the original object. Other distance measures are presented in [1].

Assume to have a function $Dist \in \mathcal{F}$, taking two constraint tuples with the same schema and returning the tuple composed by only one attribute, representing the distance between the objects represented by the constraints. By using this function, we can extend the set selection operator to deal with the condition $(dist, \epsilon)$. In particular:

$$\sigma_{(P, (dist, \epsilon))}^s(r) = \{t \mid t \in r, dist(t, P) < \epsilon\}$$

By using the previous operator, we are able to retrieve all the objects contained in r , that are similar to the image represented by P .

In a similar way, we can model the query, taking two relations R and S and returning all the pairs of objects (r, s) , such that $r \in R, s \in S, dist(r, s) < \epsilon$ in the following way:

$$\sigma_{(f_1, f_2, (dist, \epsilon))}^s(R \times \varrho_{[X|X', Y|Y']} S) = \{t \mid t \in r, dist(ext(f_1(t)), ext(f_2(t))) < \epsilon\}$$

where

$$\begin{aligned} f_1 &= \Pi_{X, Y} \\ f_2 &= \varrho_{[X'|X, Y'|Y]} \circ \Pi_{X', Y'} \\ \forall t \in R \times \varrho_{[X|X', Y|Y']} S : \alpha(f_1(t)) &= \alpha(f_2(t)) \\ \circ &\text{ identifies function composition.} \end{aligned}$$

3.2 Structure-dependent approximation

Different approximations of the same object can be used to execute different queries [6]. The same operation may have different costs when it is executed on different levels. Each approximation is related to a particular access method. The choice of the approximation(s) to use to efficiently process a query is a query processor task.

In the constraint algebra, approximated representations can be maintained by using possibly different theories or the same theory but constraints of different complexity. This implies that each approximation level corresponds to a particular query complexity, in that different classes of queries can be expressed and executed at each level.

We may assume the existence of a relation for each approximation level. The relation may contain some attributes, in order to connect each approximated representation at a particular level to the corresponding approximation at the level immediately above and/or above.

The constraint relational algebra for a constraint database modeling approximation is an extension of the algebra defined in Section 2.2, in the sense that constraints used by the queries, for example to implement the selection operator, must be modeled by the theory of the relation to which the operation refers. Moreover, different functions can be considered for each approximation level. The obtained language has the form: $(\text{GRA}(\Phi_1, \dots, \Phi_n, \mathcal{F}_1, \dots, \mathcal{F}_n))$

3.3 Execution-dependent approximation

In the following we discuss some issues related to the use of an execution-dependent approximation in a constraint database.

1. Assume to have n approximation levels, each represented by a constraint relation A_i , on a theory Φ_i . Assume that $\Phi_i \subseteq \Phi_{i+1}$. This means that all the constraints representable by using the theory Φ_i are also representable by using the theory Φ_{i+1} . Let Q be a query, that can be translated in the constraint relational algebra expression \overline{Q} (see [2] for a possible translation of some of the more common queries). Suppose that Φ_1 is the constraint theory used to specify the constraints occurring in \overline{Q} . In this case, the translation of the query in GRA deals with the lowest constraint theory.

Let A_i be the relation containing approximated images of level i . The application of a sequence of filter steps, for example by first using A_i and then A_{i+1} , is modeled by the following GRA expressions:

filter at level i : $F_1 = \overline{Q}(A_i)$
 retrieval of the objects at level $i + 1$, corresponding to the objects retrieved by the
 filter at step i : $F_2 = \sigma_{Id=Id'}^s F_1 \times \varrho_{[X/X', Y/Y', Id/Id']} A_2$

filter at level $i + 1$: $F_3 = \overline{Q}(F_2)$.

The process can be iterated.

2. If the constraint theory used to specify an image object inside Q is not Φ_1 , the filter step cannot start from the first approximation level, in that the query constraint theory is not compatible with the theory of the approximation.

Two are the possible solutions:

- (a) the filter step starts from the first approximation level whose theory allows the query representation;
 - (b) a mechanism to approximate queries on lower approximation levels is defined.
3. In both cases, another problem is the detection of an approximation level at which stopping the filter step.

4 Conclusions and future work

Constraint databases represent a promising approach to the problem of modeling and retrieving images in multimedia databases. In this paper, we have established some preliminary directions and questions to be addressed in order to completely define such approach.

Future work includes sample applications of this approach for specific types of query predicates, in particular shape and layout matching, and a comparison with other approaches. An investigation on the notion of refinement among constraint theories is also planned. Finally, we plan to investigate approximation strategies in the framework of constraint deductive databases.

References

- [1] A. Analyti and S. Christodoulakis. Multimedia Object Modeling and Content-Based Querying. In *Advanced Course - Multimedia Databases in Perspective*, University of Twente, The Netherlands, June 1995.
- [2] A. Belussi, E. Bertino, M. Bertolotto, and B. Catania. Generalized Relational Algebra: Modeling Spatial Queries in Constraint Databases. Technical Report, University of Milano, 1995. Also submitted for publication.
- [3] E. Bertino, B. Catania, and E. Ferrari. research Issues in Multimedia Query Processing. In *Advanced Course - Multimedia Databases in Perspective*, University of Twente, The Netherlands, June 1995.
- [4] A. Brodsky, J. Jaffar, and M.J. Maher. Toward Practical Constraint Databases. In *Proc. Of the 19th VLDB Conference*, pages 567–579, Dublin, Ireland, 1993.
- [5] A. Brodsky and Y. Kornatzky. The $\mathcal{L}_{\text{yriC}}$ Language: Querying Constraint Objects. In *Proc. of the ACM SIGMOD Int. Conf. on Management of Data*, 1993.
- [6] T. Brinkhoff, H.P. Kriegel, and R. Schneider. Comparison of Approximations of Complex Objects Used for Approximation-based Query Processing. In *Proc. of the IEEE Conference on Data Engineering*, pages 40–49, 1995.
- [7] A. Brodsky, C. Lassez, J.L. Lassez, and M. Maher. Separability of Polyhedra and a New Approach to Spatial Storage. In *Proc. of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, San Jose, California, 1995.
- [8] L. De Floriani, P. Marzano, and E. Puppo. Spatial Queries and Data Model. In *Spatial Information Theory: a Theoretical Basis for GIS*, A.U. Frank, I. Campari, U. Formentini, editors, volume 716 of Lectures Notes in Computer Sciences, Springer-Verlag, pages 123–138, September 1993.

- [9] M. Gargano, E. Nardelli, and M. Talamo. Abstract Data Types for the Logical Modeling of Complex Data. *Information Systems*, 16(6):565-583, 1991.
- [10] R.H. Gueting and M. Schneider. Realm-Based Spatial Data Types: The ROSE Algebra. *Ferruniversität Hagen Informatik-Report* 141, 1993. To appear in the *VLDB Journal*.
- [11] P. Kanellakis. Elements of Relational Database Theory. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, Chapter 17, 1990.
- [12] M. Koubarakis. Representation and Querying in Temporal Databases. In *Proc. of the Int. Conf. on Data Engineering*, pages 327-334, 1993.
- [13] P.C. Kanellakis, G.M. Kuper, and P.Z. Revesz. Constraint Query Languages. *Journal of Computer and System Sciences*, to appear. See also *Proc. of the 9th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 299-313, Denver, Colorado, April 1990.
- [14] J.L. Lassez. Querying Constraints. In *Proc. of the Eleventh ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 288-298, Nashville, Tennessee, April 1990.
- [15] J.A. Orenstein and A. Manola. PROBE Spatial Data Modeling and Query Processing in an Image Database Application. *IEEE Transactions on Software Engineering*, 14(5):611-629, May 1988.
- [16] J. Paredaens, J. Van den B., and D. Van G. Towards a Theory of Spatial Database Queries. In *Proc. of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 279-288, Minneapolis, Minnesota, USA, 1994.
- [17] F. P. Preparata and M.I. Shamos. *Computational Geometry - an Introduction*, Springer Verlag, New York, 1985.
- [18] M. Scholl and A. Voisard. Thematic Map Modeling. In *Proc. of the Symp. on the Design and Implementation of Large Spatial Databases*, 1989, pages 167-190.
- [19] P. Svensson. GEO-SAL: a Query Language for Spatial Data Analysis. In *Proc. of the 2nd Symp. on Advances in Spatial Databases*, pages 119-140, 1991.

Advanced Video Information System: Data Structures and Query Processing *

Sibel Adalı, Kasım S. Candan, Su-Shing Chen[†],
Kutluhan Erol, and V.S. Subrahmanian
Institute for Advanced Computer Studies
Institute for Systems Research
Department of Computer Science
University of Maryland, College Park, MD 20742.
{sibel, candan, kutluhan, vs}@cs.umd.edu, schen@nsf.gov

Abstract

In this paper, we describe how video data may be organized and structured so as to facilitate queries. We develop a formal model of video data and show how spatial data structures, suitably modified, provide an elegant way of storing such data. We develop algorithms to process various kinds of video queries and show that in most cases, the complexity of these algorithms is linear. We develop algorithms to update these video databases. A prototype system called AVIS ("Advanced Video Information System") has been designed at the University of Maryland based on these concepts.

*This research was supported by the Army Research Office under grant DAAL-03-92-G-0225, by the Air Force Office of Scientific Research under grant F49620-93-1-0065, by ARPA/Rome Labs contract Nr. F30602-93-C-0241 (Order Nr. A716), and by an NSF Young Investigator award IRI-93-57756.

[†]Su-Shing Chen is with: Information Technology & Organizations Program, National Science Foundation, 4201 Wilson Blvd, Arlington, VA 22230.

Hybrid System Methods for Distributed Multimedia Systems

Wolf Kohn
Sagent Corporation
Bellevue, Washington
e-mail: Wolf_Kohn%SAGENT@notes.worldcom.com

Anil Nerode*
Mathematical Sciences Institute

John James†
Sagent Corporation
P.O. Box 173, Fairfax Station, Virginia 22039
e-mail: 755767.77@compuserve.com

September 7, 1995

Hybrid Systems Hybrid systems are networks of interacting digital devices and continuous plants reacting to a changing environment. Our multiple agent hybrid control architecture ([KN93b], [KN93c]) is based on the notion of hybrid system state. The latter incorporates evolution models using differential or difference equations, logic constraints, and geometric constraints. The set of hybrid states of a hybrid system can be construed in a variety of ways as a differentiable (or a C^∞) manifold which we have called the carrier manifold ([KNRG95]). We have suggested that for con-

*sponsored by Army Research Office contract DAAL03-91-C-0027, SDIO contract DAAH04-93-C-O113

†sponsored by Army Research Office contract DAAL03-91-C-0027, SDIO contract DAAH04-93-C-O113

trol problems the coordinates of points of the carrier manifolds should be selected to incorporate all information about system state, control state, and environment needed to choose new values of control parameters.

Hybrid Knowledge Bases V. S. Subrahmanian and A. Nerode introduced a state spaced notion of a hybrid knowledge base. This led Subrahmanian and his students to an important and useful software implementation. He and Sherry Marcus [2, 1, 3] have subsequently developed a state space based foundation for multimedia information systems. We apply the Kohn-Nerode model of hybrid systems to extract control of synchronization and other features for distributed multimedia systems, thereby extended the scope of the theory.

Motivation Why bother? Because future will bring distributed multimedia information systems which have to synthesize multimedia information for realtime use from physically distant, dissimilar, computational platforms with different multimedia datastructures into a single environment with a shared perception of time and space for the users.

Obstacles What are the problems? Temporal and spatial synchronization of multimedia platforms into an integrated operating environment has to be modelled by large, complex, shared data structures requiring significant communication resources. Each additional platform will require additional bandwidth. The bandwidth of the connecting network will become the limiting factor for the scalability of a total system.

What does the Kohn-Nerode multiple agent hybrid control architecture offer? It promises to reduce the communications bandwidth required to achieve such shared real time perceptions based on multimedia information databases. We offer a solution to the problems encountered by those who have attempted to establish scalable, heterogenous, distributed architectures: realism, synchronization, interoperability, scalability, and validation and verification.

What kinds of uses are involved? Such systems can be used for training to enhance and supplement the skills of spatially separated teams. This has proved useful in the past for simulators for operating platforms (e.g., spacecraft, aircraft, tanks, railroad locomotives, power plant control rooms, etc.) As computational price/performance ratios have decreased with the growth of processor technology, these simulators have become extremely high fidelity representations of the physical systems, some even in the entertainment industry. Many multimedia systems exist to provide training opportunities:

few have integrated multimedia information sources; almost none have been incorporated into integrated environments.

We introduce a Kohn-Nerode “carrier manifold” corresponding to each distributed multimedia system, a state-based approach which extends the Marcus-Subrahmanian [2, 3, 1] model. There is an agent assigned to each multimedia site, contributing certain coordinates to the carrier manifold. Each agent is responsible for local synchronization and realism at its site as it receives multimedia messages from other sites. media.

Cost Functions The synchronization and realism criteria for multimedia systems in terms of a cost function for each agent. Each agent’s cost function is updated by contributions from all other agents at prescribed times. The synchronization and integration imposed is computed as the solution of a calculus of variations problem on the carrier manifold.

Our Previous Work We have previously developed a multiple agent distributed hybrid control architecture (MAHCA) for extracting control programs for systems of cooperating agents controlling distributed hybrid systems based on distributed Lagrangians and relaxed variational calculus. See ([KN92], [?], [?], [?], [?], [KN93a], [KN93b], [KN93c], [KNR93], [?], [?], [KNRY95], [GKNR95], and [KNRG95].)

We have shown how to formulate multiple cooperating autonomous software agents controlling a hybrid system and meeting dynamical and logical constraints and goals as a single distributed relaxed variational problem. We have introduced algorithms for solving the latter problem for the required controls. These algorithms stem from the measure valued calculus of variations, Bellman’s dynamic programming, the chattering lemma, the Eilenberg-Schutzenger theory of linear automaton equations, and the Lie theory of cones of controls in Lie algebras.

Applications Developed Specific applications are being developed for a variety of areas such as national distributed multi-media systems for interactive video-audio-text on demand, for manufacturing processes, for virtual enterprises, for distributed interactive simulation, for traffic control, for wireless-broadband communications networks, for battlefield command and control, for medical information systems, for medical diagnosis and health care delivery systems, for computer aided control system design, for distributed cost-benefit analysis, and very recently for compression-decompression algorithms for high definition video and video conferencing. See ([NJK94], [NJK94a], [NJK94b], [NJKD94], [NJKD94b], [NJKHA94],

[LGKNC94], [CJKNRS95]).

Models have been implemented and simulated for traffic control, multiple cell factory floors, producer consumer games, and by the Kohn-Nerode group in cooperation with the U. S. Army Picatinny Arsenal for a simulation of multiple agent control of cooperating tanks.

References

- [C35] Caratheodory, C.: *Variationsrechnung und Partielle Differentialgleichennengerster Ordnung*. Teubner, Berlin. (1935) (Dover 2nd English edition, 1982)
- [C34] Cartan, E.: *Les Espaces de Finsler*. Actualities scientifiques et industrielle 79, Exposes de geometrie II (1934)
- [E83] Ekeland, I.: *Infinite dimensional optimization and convexity*. University of Chicago Press. (1983)
- [E90] Ekeland, I.: *Convexity methods in Hamiltonian mechanics*. Springer-Verlag. (1990)
- [ET76] Ekeland, I., Temam, R.: *Convex analysis and variational problems*. North-Holland. (1976)
- [CJKNRS95] Cummings, B., James, J., Kohn, W., Nerode, A., Rimmel, J.B., Shell, K.: Distributed MAHCA Cost-Benefit Analysis. MSI Tech. Report, Cornell University. (1995)
- [GKNR95] Ge, X., Kohn, W., Nerode, A., Rimmel, J.B.: Algorithms for Chattering Approximations to Relaxed Optimal Control. MSI Tech. Report 95-1, Cornell University. (1995)
- [GNRR93] Grossman, R.L., Nerode, A., Ravn, A., Rischel, H. (eds.): *Hybrid Systems*. Springer Lecture Notes in Computer Science, Springer-Verlag, Bonn. (1993)
- [K88] Kohn, W.: A declarative theory for rational controllers. Proceedings of the 27th IEEE CDC, Vol. 1. (1988) 131-136.

- [K90] Kohn, W.: Declarative hierarchical controllers. Proc. of the Workshop on Software Tools for Distributed Intelligent Control Systems, Pacifica, CA, July 17-19 (1990) 141-163.
- [K93] Kohn, W.: Multiple agent inference in equational domains via Infinitesimal operators. Proc. of Application Specific Symbolic Techniques in High Performance Computing Environments, The Fields Institute, Oct. 17-20 (1993)
- [KN92] Kohn, W., Nerode, A.: An autonomous control theory: an overview. IEEE Symposium on Computer Aided Control System Design (March 17-19, 1992, Napa Valley, CA) (1992) 204-210
- [KN93a] Kohn, W., Nerode, A.: Models for hybrid systems: automata, topologies, controllability and observability. in [GNRR93] (1993)
- [KN93b] Kohn, W., Nerode, A.: Multiple agent autonomous control. Proceedings of the 31st IEEE CDC. (1993) 2956-2966
- [KN93c] Kohn, W., Nerode, A.: Multiple agent autonomous hybrid control systems. *Logical Methods* (Crossley, J., Rummel, J.B., Shore, R., Sweeder, M. eds.), Birkhauser. (1993)
- [KNR93] Kohn, W., Nerode, A., Rummel, J.B.: Agents in hybrid control. MSI Technical Report 93-101, Cornell University. (1993)
- [KNRG95] Kohn, W., Nerode, A., Rummel, J.B., Ge, X.: Multiple agent hybrid control: carrier manifolds and chattering approximations to optimal control. CDC94 (1994)
- [KNRY95] Kohn, W., Nerode, A., Rummel, J.B., Yakhnis, A.: Viability in hybrid systems. *J. Theoretical Computer Science.* **138** (1995) 141-168
- [KNS95] Kohn, W., Nerode, A., Subrahmanian, V.S.: Constraint logic programming: hybrid control and logic as linear programming. MSI Technical Report 93-80, Cornell University. (1993) (to appear in CLP93 1995)
- [LC17] Levi-Civita, T.: *Rendiconti del Circolo Matematico di Palermo*, fascicolo XLII. (1917)

- [LC26] Levi-Civita, T.: *The Absolute Differential Calculus*. Blackie and Sons. (1926) (Dover Reprint. 1977)
- [LNRS93] Lu, J., Nerode, A., Remmel, J.B., Subrahmanian, V.S.: Toward a theory of hybrid knowledge bases. MSI Technical Report 93-14, Cornell University. (1993)
- [LGKNC94] Lu, J., Ge, X., Kohn, W., Nerode, A., Coleman, N.: A Semi-autonomous multiagent decision model for a battlefield environment. MSI Technical Report, Oct. 1994, Cornell University. (1994)
- [1] A. Brink, S. Marcus and V.S. Subrahmanian. Heterogeneous Multimedia Reasoning. *IEEE COMPUTER*, 28, 9, pps 33-39, Sep. 1995.
- [2] S. Marcus and V.S. Subrahmanian. Multimedia Database Systems, to appear in: S. Jajodia and V.S. Subrahmanian (eds.), *Multimedia Database Systems: Issues and Research Directions*, Springer, Nov. 1995.
- [3] S. Marcus and V.S. Subrahmanian. Foundations of Multimedia Information Systems, submitted for journal publication, 1994.
- [NJK92] Nerode, A., James, J., Kohn, W.: Multiple agent declarative control architecture: A knowledge based system for reactive planning, scheduling and control in manufacturing systems. Intermetrics Report, Intermetrics, Bellevue, Wash., Nov. (1992)
- [NJK94] Nerode, A., James, J., Kohn, W.: Multiple Agent Hybrid Control Architecture: A generic open architecture for incremental construction of reactive planning and scheduling. Intermetrics Report, Intermetrics, Bellevue, Wash., June (1994)
- [NJK94a] Nerode, A., James, J., Kohn, W.: Multiple agent reactive control of distributed interactive simulations. Proc. Army Workshop on Hybrid Systems and Distributed Simulation, Feb. 28-March 1, (1994)
- [NJK94b] Nerode, A., James, J., Kohn, W.: Multiple agent reactive control of wireless distributed multimedia communications networks for the digital battlefield. Intermetrics Report, Intermetrics, Bellevue, Wash., June (1994)

- [NJKD94] Nerode, A., James, J., Kohn, W., DeClaris, N.: Intelligent integration of medical models. Proc. IEEE Conference on Systems, Man, and Cybernetics, San Antonio, 1-6 Oct. (1994)
- [NJKD94b] Nerode, A., James, J., Kohn, W., DeClaris, N.: Medical information systems via high performance computing and communications. Proc. IEEE Biomedical Engineering Symposium, Baltimore, MD, Nov. (1994)
- [NJKHA94] Nerode, A., James, J., Kohn, W., Harbison, K., Agrawala, A.: A hybrid systems approach to computer aided control system design. Proc. Joint Symposium on Computer Aided Control System Design. Tucson AZ 7-9 March (1994)
- [NLS95] Nerode, A., Lu, J., Subrahmanian, V.S.: Hybrid knowledge bases. IEEE Trans. on Knowledge and Data Engineering. (to appear)
- [Y80] Young, L.C.: *Optimal Control Theory*. Chelsea Pub. Co. N.Y. (1980)

Benchmarking Digital Video

Richard Gerber and Ladan Gharai
Department of Computer Science
University of Maryland
College Park, MD 20742
{rich, ladan}@cs.umd.edu

Abstract

Digital video applications can often push a computer's resources to their limit. They require massive amounts of storage, high IO transfer rates, and fast display refresh times. And if software is involved in the decompression process, the CPU will often end up over-utilized. Resource demands have a direct effect on the quality of the delivered video; this results in a complex "load-balancing" problem, which must be solved with both quantitative and qualitative metrics.

In this paper we describe our experiments on media applications, specifically concentrating on the tradeoff analysis involved in tuning video systems. We first postulate a set of hypotheses, and then we describe the controlled set of 240 tests we ran to test them. Our observations confirmed that achieving smooth playback is mainly a problem of coordinating an operating system to the properties of the media.

The first 120 test runs were drawn from a series of 60 videos, which we generated with our own Hi8 equipment. Each test video uniquely instantiated the following variables: *compression type*, *frame size*, *digitized rate*, *spatial quality* and *keyframe distribution*. The tests were carried out on two Apple Macintosh platforms: at the lower end a Quadra 950, and at the higher end, a Power PC 7100/80. Our quantitative metrics included average playback rate, as well as the rate's variance over one-second intervals.

This paper contains the results of these experiments, as well as our analysis of each variable's importance. The first set unveiled several anomalous latencies, which could not be explained by any of the variables. Thus we ran additional 120 tests, whose analysis led us to conclude that even on a lower-end computer, a software-only solution is sufficient for good quality video playback – provided that the operating system is tuned accordingly.

Multimedia Information Systems

Sherry Marcus
21 Century Technologies
1903 Ware Road
Falls Church, VA 22043
sem@cais.com

Abstract

Although numerous multimedia systems exist in the commercial market today, relatively little work has been done on developing the mathematical foundations of multimedia technology. Marcus and Subrahmanian (1) have taken some initial steps toward the development of a theoretical basis for multimedia information systems. They define a mathematical model of a media-instance. A media-instance may be thought of as "glue" residing on top of a specific physical media-representation (such as video, audio, documents, etc.) Using this "glue", it is possible to define a general purpose logical query language to query multimedia data. This glue consists of a set of "states" (e.g. video frames, audio tracks, etc.) and "features", together with relationships between states and/or features. A structured multimedia database system imposes a certain mathematical structure on the set of features/states. Using this notion of a structure, they are able to define indexing structures for processing queries, methods to relax queries when answers do not exist to those queries, as well as sound, complete and terminating procedures to answer such queries (and their relaxations, when appropriate). Using the Marcus and Subrahmanian (1) work on multimedia database in tegration systems, we show how their logical based query language can be redefined as an SQL based query language. As there are numerous commercial SQL database systems, a wide and diverse population of users may access the Marcus and Subrahmanian work using their SQL interface. Also, there has been a great deal of study of query optimization of SQL based languages (2). Such optimizations can be applied to "SQL version" of the Marcus and Subrahmanian system.

References

- 1) S. Marcus and V.S. Subrahmanian. (1993) *Multimedia Database Systems*, submitted for publication.
- 2) A. Silberschatz, M. Stonebraker and J. D. Ullman. (1991) *Database Systems: Achievements and Opportunities*, Comm. of the ACM, 34, 10, pps 110-120.

Integrating a Scalable Rapid Reasoning System with a Database System

Extended Abstract

Workshop on Principles of Multimedia Information Systems

Lokendra Shastri¹

International Computer Science Institute

1947 Center Street

Berkeley, CA 94704

shastri@icsi.berkeley.edu

1 Introduction

In past work we have shown that a limited class of first-order inference can be performed rapidly even when dealing with very large knowledge bases (KBs). This abstract outlines a proposal for integrating the result of our work on a scalable and efficient knowledge representation and inference system with a relational database system. Since the intractability of inference is a limiting factor in the effectiveness of large deductive databases and knowledge based systems, an integration of the inferential capabilities of our system and the full functionality of existing DB systems could lead to a flexible, expressive, and efficient system for accessing large knowledge/data bases. In what follows, we briefly describe the results of our work on rapid inference and the proposed integration.

2 Reflexive reasoning

We have shown that a class of first-order inferences can be performed by a parallel network model (see below) in time proportional to the *depth* of inference and using networks whose size is only *linear* in the size of the underlying KB (Ajjanagadde & Shastri 1991; Shastri & Ajjanagadde 1993; Shastri 1993; Mani & Shastri 1993). We have referred to this class of inference as *reflexive reasoning*. In this section we give a brief description of the class of inference and the computational model.

2.1 Form of rules, facts, and queries

We refer to first-order sentences of the form:

$$\exists x_1: X_1, \dots, x_p: X_p \forall y_1, \dots, y_r, z_1: Z_1, \dots, z_s: Z_s [P_1(\dots) \wedge \dots \wedge P_n(\dots) \Rightarrow \exists u_1, \dots, u_t Q(\dots)]$$

¹This work was supported in part by ONR grant N00014-93-1-1149 and NSF resource grant CCR930001N.

as rules. An argument of P_i can be a constant or any of the x_i s, y_i s, or z_i s. An argument of Q can be a constant or any of the x_i s, y_i s, z_i s, or u_i s. X_1, \dots, X_p and Z_1, \dots, Z_s are types and specify restrictions on the bindings of x_i s and z_i s, respectively. Although a type restriction can be encoded by simply adding a unary antecedent to a rule, we treat types differently because we use a specialized representation for supertype and subtype relations (as well as other transitive relations defined over terms), in order to support the rapid computation of transitive closure.

Facts are partial or complete instantiation of predicates of the form:

$$\exists x_1, \dots, x_r, y_1:Y_1, \dots, y_s:Y_s \forall z_1:Z_1, \dots, z_t:Z_t P(\dots)$$

where Y_i and Z_i are types, and each argument of P is either a constant or a variable. It is assumed that existentially quantified variables are distinct. Queries, have the same form as facts.

2.2 Reflexive queries

Let us refer to a variable that occurs in multiple argument positions in the antecedent of a rule as a *pivotal* variable. Let us call a derivation of a query Q obtained via backward chaining *threaded*, if all pivotal variables occurring in the derivation get bound as a consequence of bindings introduced in Q . Observe that a threaded derivation can only involve rules wherein each pivotal variable also occurs in the consequent of the rule.

A *reflexive* query is any query for which there exists a threaded proof. It can be shown that a *yes* answer to a reflexive *yes-no* query Q can be obtained in time proportional to $|In|^{V+1}d$, where: $|Input|$ is the number of *distinct* bindings specified in Q , V is the maximum arity of predicates in KB, and d equals the depth of the most shallow derivation of Q . Observe that V can be treated as a constant and hence, the time required to answer a reflexive query is *independent* of $|KB|$ and is only polynomial in $|Input|$. Answers to *wh*-queries can also be computed in time proportional to $|Input|^{V+1}D$, except that $|Input|$ now equals the arity of the query predicate.

2.3 Source of efficiency

The effectiveness of inference in our model can be traced to three factors: (i) The representation of rules in the form of an explicit inferential dependency graph, (ii) the compact encoding of variable bindings, and (iii) restriction on the form of rules.

The inferential dependency graph is the graph obtained by representing each n -ary predicate by a cluster of n argument nodes and two control nodes, and a rule by links between the clusters of consequent and antecedent predicates. The links associated with a rule explicitly encode the correspondence between the arguments of consequent and antecedent predicates specified in the rule. This representation of predicates and rules supports efficient rule application and propagation of variable bindings during inference. In particular, the firing of a rule can be realized by simply propagating a compact message — specifying variable bindings — from the consequent predicate to the antecedent predicates. The size of this message need only be $V \log V$ bits where V is the maximum arity of predicates in the $|KB|$. For many practical applications, a message size of two words (integers) would suffice.

The model exhibits OR-parallelism and independent AND-parallelism. The requirement that derivations be threaded ensures that derivations will not require the computation of *joins* and also obviates the need for dependent AND-parallelism.

2.4 The computational model

Our computational model is a network of nodes connected via weighted links. The model makes use of three node types:

ρ -btu nodes: A ρ -btu node with threshold n becomes active upon receiving n synchronous inputs. In particular, a ρ -btu node B with threshold 1 receiving a periodic pulse train from a ρ -btu node A will become active and produce a periodic pulse train that is *in-phase* with the pulse train it receives from A .

τ -and nodes: A τ -and node with a threshold of 1 becomes active on receiving a pulse train of width $\geq \pi_{max}$. Thus a τ -and node behaves like a *temporal and* node. Upon becoming active, such a node produces an output pulse similar to the input pulse. A threshold, n , associated with a τ -and node indicates that the node will fire only if it receives n or more synchronous pulses.

τ -or node: A τ -or node with threshold n becomes active on receiving n or more inputs during an interval π_{max} . Upon becoming active, a τ -or node produces an output pulse of width π_{max} . Thus a τ -or node behaves like a *temporal or* node.

The model also makes use of *inhibitory modifiers*—a pulse propagating along an inhibitory modifier will block a synchronous pulse propagating along the link it impinges upon. Unless otherwise specified, the threshold of a node is assumed to be 1.

2.5 Overview of the mapping and inference

We briefly describe how rules and facts are encoded and inference performed by the model using a simple example. We suppress details pertaining to the mapping of multiple antecedent rules, the type hierarchy, and the representation of multiple predicate instantiations. A detailed description can be found in (Shastri & Ajjanagadde 1993) and (Mani & Shastri 1993). Figure 1 illustrates how long-term knowledge is encoded in the rule-based reasoning system. The network encodes the following rules and facts: $\forall x,y,z [give(x,y,z) \Rightarrow own(y,z)]$, $\forall x,y [buy(x,y) \Rightarrow own(x,y)]$, $\forall x,y [own(x,y) \Rightarrow can-sell(x,y)]$, $give(John,Mary,Book1)$, $buy(John,x)$, and $own(Mary,Ball1)$.

Each entity in the domain is encoded by a ρ -btu node. An n -ary predicate P is encoded by a pair of τ -and nodes and n ρ -btu nodes, one for each of its n arguments. In Figure 1, ρ -btu nodes are depicted as circles while the τ -and nodes are depicted as pentagons. One of the τ -and nodes in a predicate cluster is referred to as the *enabler* and labeled $e:P$, while the other is referred to as the *collector* and labeled $c:P$. In Figure 1 *enablers* point upward while *collectors* point downward. The *enabler* $e:P$ becomes active whenever the system is being queried about P . On the other hand, the system activates the *collector* $c:P$ whenever the system wants to assert that the current dynamic bindings of the arguments of P follow from the knowledge encoded in the system. A rule is encoded by connecting the *collector* of the antecedent predicate to the *collector* of the consequent predicate, the *enabler* of the consequent predicate to the *enabler* of the antecedent predicate,

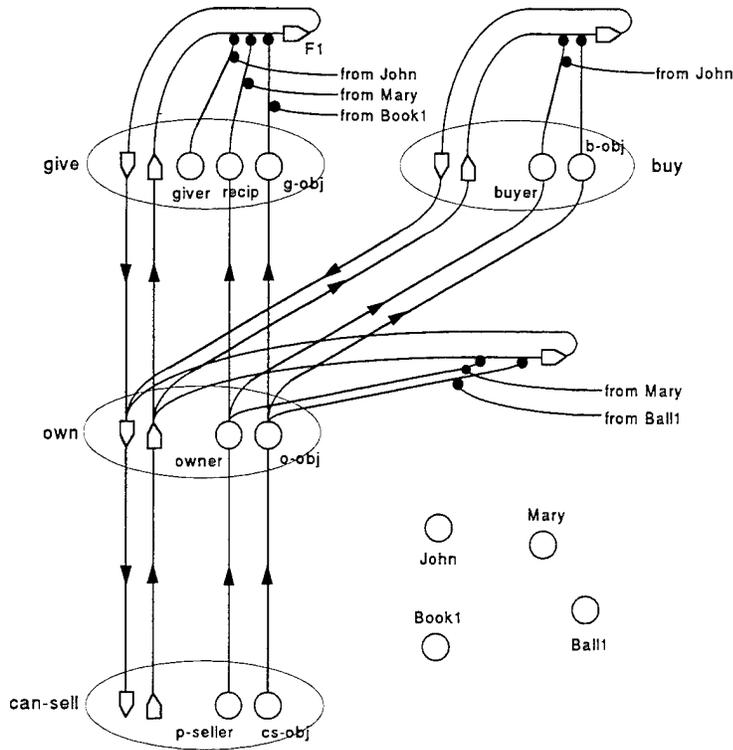


Figure 1: An example encoding of rules and facts.

and the arguments of the consequent predicate to the arguments of the antecedent predicate in accordance with the correspondence between these arguments specified in the rule. A fact is encoded using a τ -and node that receives an input from the enabler of the associated predicate. This input is modified by inhibitory modifiers from the argument nodes of the associated predicate. If an argument is bound to an entity in the fact then the modifier from such an argument node is in turn modified by an inhibitory modifier from the appropriate entity node. The output of the τ -and node is connected to the collector of the associated predicate. Figure 1 shows the encoding of the facts $give(John, Mary, Book1)$ and $buy(John, x)$. The encoding of more complex rules also makes use of the τ -or nodes mentioned above.

The Inference Process. Posing a query to the system involves specifying the query predicate and its argument bindings. The query predicate is specified by activating its *enabler*. Argument bindings are specified by activating each entity, and the argument nodes bound to that entity, with a synchronous (and periodic) pulse train. Thus all arguments bound to the same entity start firing in-phase. At the same time each distinct entity is assigned a distinct phase. (Each phase is a non-overlapping time interval within a period of oscillation).

Consider the query $can-sell(Mary, Book1)?$ (i.e., Can Mary sell Book1?) The query is posed by (i) Activating the *enabler* $e:can-sell$; (ii) Activating *Mary* and *p-seller* in the same phase (say, ρ_1), and (iii) Activating *Book1* and *cs-obj* in some other phase (say, ρ_2). As a result of these inputs, *Mary* and *p-seller* fire synchronously in-phase ρ_1 of every period of oscillation, while *Book1* and *cs-obj* fire synchronously in phase ρ_2 . See Figure 2. The activation from the *can-sell* predicate propagates to the *own*, *give* and *buy* predicates

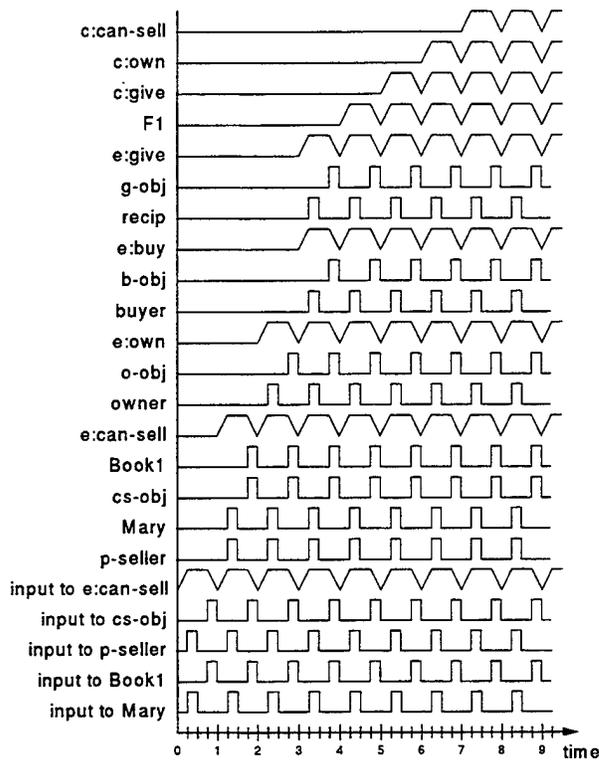


Figure 2: Activation trace for the query *can-sell(Mary, Book1)?*.

via the links encoding the rules. Eventually, as shown in Figure 2, *Mary*, *p-seller*, *owner*, *buyer* and *recip* will all be active in phase ρ_1 , while *Book1*, *cs-obj*, *o-obj*, *g-obj* and *b-obj* would be active in phase ρ_2 . The activation of *e:can-sell* causes the enablers of all other predicates to go active. In effect, the system is asking itself three more queries—*own(Mary, Book1)?*, *give(x, Mary, Book1)?* (i.e., Did *someone* give Mary Book1?), and *buy(Mary, Book1)?*. The τ -and node F1, associated with the fact *give(John, Mary, Book1)* becomes active as a result of the uninterrupted activation it receives from *e:give*, thereby answering *give(x, Mary, Book1)?* affirmatively. The activation from F1 spreads downward to *c:give*, *c:own* and *c:can-sell*. Activation of *c:can-sell* signals an affirmative answer to the original query *can-sell(Mary, Book1)?*.

3 Mapping the model to real machines

The massively parallel network model outlined above assumes hardwired connections and high fan-in and fan-out. In other words, it assumes constant communication costs. Clearly, this assumption does not hold for real computers. However, we believed that the capabilities of existing high performance platforms would make it feasible to map our abstract model to a real machine and obtain rapid response to reflexive queries. This belief was based in part on the excellent match between the inter-node communication requirements of our model and the active message facility provided by the CM-5 for low-latency interprocessor communication of short messages. Indeed, the amount of information that needs to be communicated among predicate clusters during rule-firing was found to match the information capacity of active messages in the CM-5.

The knowledge representation and reasoning model has been mapped onto the CM-5 (Mani & Shastri 1994; Mani 1995). The resulting system can encode a KB with over 500,000 (randomly generated) rules and facts, and yet respond to a range of queries requiring derivations of depth five in under 250 milliseconds. Even queries with derivation depths of eight are answered in well under a second. We have also encoded WordNet — a large real-world lexical “thesaurus” which encodes relations such as hypernym, hyponym, synonym, and holonym among lexical item — on the CM-5 and obtained response times ranging from a few to a few hundred milliseconds for typical WordNet queries. For example, a query like “What are the subconcepts of *Bird*?” (which activates a large number of entities) requires 290 milliseconds while a simpler query like “What are the synonyms of *Bird*?” takes only 4 milliseconds.

3.1 An overview of the KB system implementation on the CM-5

The individual processing elements on the CM-5 are relatively powerful processors. Hence, a subnetwork in the parallel network model can be implemented on a CM-5 processor using appropriate data structures and associated procedures without necessarily mimicking the detailed behavior of individual nodes and links in the subnetwork. This lead us to adopt a *knowledge-level* mapping of the model onto the CM-5 (rather than a node level mapping). At this level of granularity, KB elements like predicates, entities, facts, rules and *is-a* relations form the primitives of the mapping. The CM-5 implementation provides an integrated environment for encoding very large KBs and using them to perform real-time inference and retrieval. This includes:

- A parser for accepting KB items expressed in a human readable input language whose syntax is similar to that of first-order logic.
- A pre-processor for mapping a KB onto the CM-5. This involves mapping the KB to an inferential dependency graph whose structure is analogous to that of the parallel network model and partitioning this graph among the processors of the CM-5.
- A reasoning algorithm for answering queries. This runs asynchronously on the CM-5 processors.
- An interface for adding or deleting KB items from a KB already loaded on the CM-5.
- Procedures for collecting a number of statistics about the KB and the reasoning process.
- A utility for generating large pseudo-random KBs given a specification of broad structural constraints on the KB.
- A number of user friendly tools for analyzing and visualizing the KB and the statistics gathered during query answering.

4 Proposed integration of KB and DB system

The proposed integration of our knowledge representation and reasoning system with a DB system intends to leverage the full functionality of the DB system. This includes the ability to execute all the standard relational operations, perform query optimization, and support parallel query execution.

In the integrated system, all "facts" will reside in the DB and the KB component will consist only of rules, the type hierarchy, and transitive terminological relations. Furthermore, rules will not have to satisfy the constraint on pivotal variables since the DB component would be capable of computing joins.

We propose to use the KB component of the integrated system to encode two sorts of knowledge: (i) a set of rules that capture salient inferential dependencies among the relations in the underlying DB system and (ii) terminological relations such as synonym, hypernym, and hyponym, among terms and concepts that appear in the DB or user queries.

The KB component will make use of its knowledge and inferential power in the following ways. First, it will enable the integrated system to give the user considerable terminological latitude and flexibility in framing queries. This flexibility is expected to result from the KB system's ability to use terminological relationships to map terms occurring in a user query into canonical terms interpretable by the query processor of the underlying DB system.

Second, the KB component of the integrated system would enhance the retrieval efficiency of the DB system. The KB would carry out rapid inference over rules that capture inferential dependencies in the underlying DB to map a given user query into a large number of alternate (but equivalent) queries. An answer to any of the alternatives would constitute an answer to the initial query. The mapping of queries to a set of alternate (but equivalent) queries could lead to improved retrieval efficiency since a query optimizer can evaluate the alternate queries and pick the one with the least expected cost. Even greater speedup would be obtained if the underlying DB supports parallel processing of queries. In this case, the query optimizer might select several promising alternatives and execute them simultaneously.

In order to illustrate the above, consider a simple example where the DB is a relational DB (RDB) containing the binary relations: P, Q, R, S and T. Assume that the KB component contains the rules:

- $\forall x, y; T(x, y) \Rightarrow P(x, y)$
- $\forall x, y; P(x, y) \& Q(y, x) \Rightarrow R(x, y)$
- $\forall x, y; S(y, x) \Rightarrow R(x, y)$

Now given the query $R(a, b)$ the KB can rapidly map the query to the following set of queries: (i) $R(a, b)$, (ii) $S(b, a)$, (iii) $P(a, b) \& Q(b, a)$, and (iv) $T(a, b) \& Q(b, a)$. An answer to any of these queries constitutes an answer to the original query. Now each of these queries would be passed on to the query optimizer which can select the one with the least expected cost. Furthermore, if the underlying DB supports parallelism, then all or a subset of the above queries (chosen by the query optimizer) could be processed simultaneously.

In addition to the potential increase in efficiency, the KB can extend the set the queries that can be answered by the underlying RDB. The integrated system is capable of retrieving not just tuples in the underlying DB but also any tuple that lies in the closure under reflexive inference of the items in the RDB and the KB. With reference to above rules, it would be possible to answer the query $R(a, b)$? even if $R(a, b)$ does not exist in the RDB, as long as $S(b, a)$ does. Furthermore, consider the situation where the relations P and R do not exist in the RDB. If however, the KB contains the above rules, the integrated system would be able to map a query such as $R(a, b)$ to two alternate queries: $S(b, a)$ and $T(a, b) \& Q(b, a)$, and hence answer them — even though the query predicate does not exist in the RDB.

References

- Ajjanagadde, V. and Shastri, L. (1991) Rules and variables in neural nets. *Neural Computation*. 3, 121-134, 1991.
- Mani, D.R. and Shastri, L. (1993) Reflexive Reasoning with Multiple-Instantiation in a Connectionist Reasoning System with a Typed Hierarchy. *Connection Science*, Vol. 5, No. 3 & 4, 205-242. 1993.
- Mani, D.R. and Shastri, L. (1994) Massively parallel real-time reasoning with very large knowledge bases: An interim report. Technical Report TR-94-031, International Computer Science Institute.
- Mani, D.R. (1995) The design and implementation of massively parallel knowledge representation and reasoning systems. Forthcoming Ph.D. dissertation, University of Pennsylvania.
- Shastri, L. and Ajjanagadde, V. (1993) From simple associations to systematic reasoning. *Behavioral and Brain Sciences* Vol. 16, No. 3, 417-494, 1993.
- Shastri, L. (1993) A Computational Model of Tractable Reasoning - Taking Inspiration from Cognition. *Proceedings of IJCAI-93*, France, August-September 1993. pp. 202-207.

On storage and retrieval of generalized spatial data

Naphtali Rishé

High-performance Database Research Center
School of Computer Science
Florida International University
University Park, Miami, FL 33199
Telephone: (305) 348-2025, 348-2744
FAX: (305)-348-3549; E-mail: rishen@fiu.edu

We are developing a prototype high-performance semantic spatial database management system. One type of data in this system is a generalized spatial function — a function from a Cartesian product of several continuous and/or discrete domains into a Cartesian product of powersets of continuous domains and/or discrete domains and/or sets of semantic facts. For example, ocean temperature is a function f from $X*Y*Z*T*O$, where X , Y , Z , and T are the space-time continuum and O is a discrete set of observation stations that reported measurements. $f(x,y,z,t,o)=(s,i)$, where s is a segment of temperatures (e.g., 50 degrees plus or minus 0.01 degrees) and i is a set of semantic facts. Another example is remote photography of ocean color by the SeaWiFS satellite.

The spectrum of problems we have addressed concerning this data type includes:

1. Highly-efficient basic queries, including "inverse" queries (e.g., "Where is the temperature of about 70 degrees?")
2. Compact lossless storage
3. Compact lossy storage, particularly by approximating function values.
4. Efficient complex queries
5. Load balancing between processors and storage units
6. Visual presentation of query results as animated movies. (E.g., if the output of a query is a three-dimensional function, any two of the dimensions are mapped on the screen and the third presented as a frame sequence.)

CONTENT-BASED RETRIEVAL OF PICTURES AND VIDEOS

A. P. Sistla Clement Yu

June 30, 1995

Extended Abstract

We sketch what we have accomplished in Section 1 and what we plan to do in Section 2.

1 Summary of Our Results

In our earlier works (see [ATY95, SYH94, SYLL95, SY95]) we have developed an early version of a content based picture retrieval system. In this section we summarize this work.

1.1 Picture Representation

We make use of E-R/semantic-net model to represent contents of pictures. The contents of a picture is a collection of objects related by some associations. Thus, a picture can be represented as follows.

entities: objects identifiable in a picture.

attributes: properties (color, size etc) that qualify or characterize objects.

relationships: associations among the identified objects.

For example, consider a picture containing a tall man shaking hands with a woman and is to the left of the woman. The entities/objects in the picture are the man and the woman. The value of the height attribute for the man is “tall”. There are two relationships among the objects in this picture—“left-of” and “shaking hands”.

One of the relationships “left-of” is a spatial relationship, while the other “shaking hands” is an action relationship.

We group the relationships of objects into the following two types:

- **Action Relationships:** Relationships which describe an action are classified under this group. In the above example “shaking hands” is an action relationship.
- **Spatial Relationships:** The type of relationships that state the relative positions of two entities are grouped under type *spatial*. Examples are *above*, *in-front-of*, *below*, etc. In the above example *left-of* is a spatial relationship.

Another example is the X-ray image of the lungs of a patient. The image indicates a large tumor inside the left lung. The entities in this picture are the two lungs and the tumor. The “size” attribute value of the tumor is “large”. The spatial relationship is *inside*.

1.2 User Interface

We have developed an iconic interface that guides the user step by step in specifying the contents of the picture that the user has in mind. The interface has features to identify the objects (entities), their characteristics (attribute values), and the associations (relationships) among the objects.

The user interface is organized into several distinct parts, which take the form of separate windows or frames. The first frame is the *picture representation area* which displays the description-in-progress. The picture representation area shows a symbolic version of the picture, using icons and text labels. The second major frame is the *icon palette*, which contains icons representing objects to be inserted into the picture.

Upon startup, the interface displays the picture representation area and the icon palette. The palette contains icons which represent several classes of entities, which have been chosen based on the range of object types which may occur in real pictures. Examples of these are *man, woman, boy, girl, baby, building, thing* (generic entity), *plant*, and *animal*. The palette can be changed to suit the needs of application domain. The user is prompted to choose one or more icons with the pointing device and drag them into the picture representation area. After moving a new icon into the picture representation area, the user is immediately prompted to specify the values of a set of attributes such as name, age, etc. This consists of first selecting a set of attributes. Whenever an attribute is selected, a dialog box for the attribute is displayed. The user either chooses a value among a given set of values or inputs a value from the keyboard.

After this, the user can specify the relationships among the objects. To do this, the user is guided through a sequence of input frames which elicit the relationship description, one piece of information at a time. The goal in using this method is to gain both ease of use and lack of ambiguity in the final description. For each relationship, the user is asked to classify the relationship as either an *action* or as a *spatial*, and as either as a *directional* or as a *mutual* relationship. A mutual action is simply a symmetric action. For example *shaking hands* is a mutual relationship. A directional relationship has a subject entity and an object entity, and is anti-symmetric. For example *shooting* is a directional action relationship. In the case of action relationship, the user is prompted for a name describing the action, such as *shooting* or *chasing*. In the case of a spatial relationship, the user is given a fixed list of names from which to choose, and there is no need for the user to specify whether it is a mutual or a directional relationship.

We assume that there is a database containing the pictures. We also assume that each picture is associated with some meta-data describing the contents of the picture. This meta-data contains information about the objects in the picture, their properties and the relationships among them. For example, consider a picture containing a tall man shaking hands with a slender woman and is to the left of the woman. The meta-data about this picture identifies two objects, man and woman with attribute values "tall" and "slender" respectively, and the spatial relationship *left_of* and the non-spatial relationship *hand_shaking*. We assume that this meta-data is generated a priori (possibly, by image analysis algorithms, or manually, or by a combination of both), and is stored in a separate database. This meta-data will be used by the query processing mechanism in determining the pictures that need to be retrieved in response to a query. The meta-data facilitates efficient query processing, i.e. it avoids the invocation of the expensive image analysis algorithms each time a query is processed.

1.3 Computation of Similarities

Our system uses similarity based retrieval. This involves computation of a similarity value with each picture that denotes how closely the picture matches the query, and retrieving those pictures with the highest similarity values. Such retrievals are needed when the user cannot provide a precise

specification of what he/she wants. Even if the user can precisely specify his/her requirements, there may not be any pictures in the database that exactly match with the user's query, and in this case the user may want the closest matches. It is to be noted that, when we use similarity based retrieval, any picture that matches exactly with the query will be assigned the maximum similarity value, and will be automatically retrieved first.

In our system the similarity of a picture P with respect to a query Q is computed as the sum of three numbers called *object similarities*, *non-spatial similarities*, and *spatial similarities*. The object similarities denote how closely the objects in Q match with the objects in P . The non-spatial and spatial similarities are based on matching of the non-spatial and spatial relationships.

The object similarity of an object in P with respect to an object Q is computed based on their types (e.g. "man" is an object type) and their attribute values. Identical types yield higher similarity than types related by an IS-A relationship or through thesaurus. Identical attribute values yield higher similarity than attribute values that are related using fuzzy matching/citeYWLY95. For example, the value "very young" for the age attribute fuzzy matches the value "young".

The non-spatial similarity of two relationships (one in Q and another in P) is computed by matching the relationship names exactly or by an electronic thesaurus, and by matching the two sets of objects involved in the two relationships. The two sets of objects are matched approximately yielding different similarities based on the degree of matching (see [ATY95] for details).

Spatial Similarities, Deduction and Reduction

When computing the spatial similarity value of a picture with respect to a query, in general, we need to check if the picture satisfies any of the spatial relationships which are *implied* by the user's query, in addition to the explicitly stated relationships. For this reason, it becomes important to compute all the implied spatial relationships of the query; we call this problem as the *deduction* problem. On the other hand, some of the spatial relationships of the user's query which are satisfied by the picture may be redundant, i.e. they may be implied by other satisfied relationships; such redundant relationships should be given lower weights or they should not be considered in the computation of the similarity value of the picture. For this reason, it becomes essential to compute the reduction of a set F of spatial relationships which is the smallest subset of F that implies all the relationships in F .

In our earlier work [SYH94], we presented a set of rules for deducing new spatial relationships from a given set F of existing spatial relationships. We showed these rules to be *sound* and *complete*, i.e. every deduced relationship is implied by relationships in F , and every relationship implied by the relationships in F can be deduced using the rules. We had also given [SYLL95] efficient algorithms for directly computing the set of relationships implied by F , and the reduction of F .

The spatial similarities are defined as follows. For the query Q , we use the deduction algorithm to compute $ded(Q)$, which is the set of relationships implied by Q . After this, the set of relationships in $ded(Q)$ that are satisfied by a given picture P are identified. The reduction of this set is computed, which automatically eliminates redundant relationships. The relationships in this reduction are used in defining the spatial similarity of P with respect to Q .

Indices On Spatial Relationships

When the number of pictures in the database is very large, it becomes impractical to explicitly examine each picture to compute its spatial similarity value. We avoid this problem by employing indices on the spatial relationships. In our earlier work [SYLL95], we presented two efficient methods for computing the similarity values of those pictures that satisfy at least one of the spatial relationships in the query. Furthermore, these methods employ the deduction and reduction algorithms in the computation of the similarity values. We suppose that the user specifies an integer u and would like to retrieve the u pictures having the highest similarity values. The first method computes the exact spatial similarity values and retrieves the u pictures having the highest spatial similarity values; it is

very efficient when the number of spatial relationships given in the query is small.

The second method estimates the spatial similarity values, and retrieves the meta data of the *cu* pictures having the highest estimated spatial similarity values for some constant $c > 1$; it then computes the actual spatial similarity values of these and retrieves the *u* pictures, among these *cu* pictures, having the highest spatial similarity values; this method is efficient if the number of relationships is large. Experimental results for the second method show that for a value of $c = 3$, in all practical cases, the *u* pictures with the highest spatial similarity values among all the pictures in the database are contained in the *cu* pictures having the highest estimated spatial similarity values, and will therefore be retrieved.

The method for similarity computation assigns weights to the object types, attribute values, spatial and non-spatial relationships depending on their frequencies of occurrences in the database of pictures and using the inverse document frequency method [Salt89].

An initial prototype has been built. We have carried out preliminary experiments with several users on pictures extracted from TIME magazine and some travel agency brochures. The number of pictures in the database is less than 500. The experimental results show that, in most cases, the desired pictures are present among the ten pictures having the highest similarities. Thus, it is sufficient to retrieve the top ten pictures for each query. The prototype is being licensed by the University of Illinois at Chicago to an external company.

2 Summary of Proposed Research

Objective: Our plan is to conduct research and develop a system which retrieves pictures and videos accurately and efficiently based on their contents. Initially, the system will be for picture retrieval; and this will be extended for video retrieval. In the proposed system, the end users need not be familiar with any query language and need not be experienced in the use of computers. The construction of the system will be based on extensions of our earlier works [SYH94, SYLL95, ATY95, SY95].

The major research issues include— (i) Development of a user interface for specification of queries on pictorial and video databases (ii) Handling of Spatial Relationships, (iii) Similarity Computations and Accuracy Enhancements, (iv) Efficiency, (v) Handling of Temporal Conditions and Active Database component.

A user friendly visual query tool to facilitate intuitive querying by non-expert users will be developed. For picture retrieval, the user query will be represented by a symbolic picture. This symbolic picture is specified by objects (represented by standard icons) in the picture and the different relationships among these objects. Furthermore, additional conditions may be specified on the objects. We also employ the notion of a sub-picture to specify the contents of the picture hierarchically at different levels of detail. For example, in a picture containing a house, the user would be able to use a subpicture to specify the contents inside the house. Subpictures may also be used to specify contents of compound objects.

In case of queries on video databases, we plan to use the notion of primitive scenes. A primitive scene represents a logical scene which is a short sequence of frames. For the purpose of retrieval, a primitive scene is specified as a single picture using objects, relationships and actions; this is similar to the specification of a query in picture retrieval which makes it easier for users. It is to be noted that an action in a primitive scene usually involves more than one picture. A primitive scene involving an action is recognized by certain properties of sequences of pictures. For example, a primitive scene in which John Wayne shoots a bandit, may be recognized by a picture in which John Wayne holds a gun, followed shortly by another picture in which the bandit is lying on the ground. Pictures

representing such primitive scenes can be translated into temporal logic formulas [MP92] that specify the properties of sequences of pictures that correspond to the primitive scene. These temporal formulas can be directly evaluated on the meta-data denoting the contents of the various images in the video. A complex video query will be specified by composing primitive scenes.

In addition to the graphical user interfaces, as described above, we will also develop formal query languages. We plan to provide translations from the graphical language into the formal languages and study the expressive power of the graphical language. For example, we plan to use a first order query language or a modal logic for specification of pictorial queries, and a combination of temporal logic and the first order language for specifying queries on the video database. The formal query languages provide a precise semantics for the informal pictorial based languages. We plan to implement retrieval of queries specified in the pictorial query language as well as the formal query language. The formal language can be used as a query language by application programmers.

The proposed system will provide the feature of "incremental querying" where the user will be allowed to modify the query after observing some of the retrieved pictures/videos. The modifications will be carried out explicitly by the user, or automatically by the system after receiving the user's feedback on the retrieved pictures.

Our proposed system will be similarity based. Such similarity based retrievals of pictures consists of computing a similarity value with each picture that denotes how closely the picture matches the query, and retrieving those pictures with the highest similarity values. Such retrievals are needed when the user cannot provide a precise specification of what he/she wants. Even if the user can precisely specify his/her requirements, there may not be any pictures in the database that exactly match with the user's query, and in this case the user may want the closest matches. It is to be noted that, when we use similarity based retrieval, any picture that matches exactly with the query will be assigned the maximum similarity value, and will automatically be retrieved first.

In many applications, one is interested in retrieving pictures in which the objects satisfy some spatial relationships. For example, one may be interested in retrieving all pictures in which Colin Powell is to the left of President Clinton. It is non-trivial to handle the spatial relationships such as *left-of*, *inside* etc. Some of the important spatial relationships that we will consider are *left_of*, *above*, *in-front-of*, *inside*, *outside*, *overlaps*, *between* etc. There is no single well accepted definition for each of these relationships when the objects can be arbitrarily shaped. We will investigate various possible definitions for different classes of applications. For example, we may consider object *A* to be left of object *B* if every point in *A* is to the left of every point in *B* in one application, while in a different application the definition may simply mean that the center of gravity of *A* is to the left of that of *B*.

We will also investigate two important problems, deduction and reduction, associated with spatial relationships. The deduction problem is to compute all spatial relationships implied by a given set, while the reduction problem is to compute a minimum set of relationships that imply a given set. Both these problems are of much importance in similarity based retrieval of pictures (see [SYLL95]). In some applications the above spatial relationships will not be sufficient. For example, one may be interested in pictures where tanks are moving in a particular formation. The formation may not be easily described in words, but can be drawn with ease. We intend to examine the specification and handling of such non-typical spatial operators.

Apart from qualitative spatial relationships, we also plan to investigate quantitative spatial relationships. Using such relationships, one would be able to specify distances between objects in addition to the directions.

We will investigate various possible similarity functions and their effectiveness for the retrieval of pictures as well as videos, and develop algorithms for computing the similarity values of the pictures. Similarity computations involving compound objects will be done recursively.

One possible straightforward way of computing similarity values is to explicitly examine the meta-data associated with each picture/video. However, this method can be very inefficient and impractical when the number of pictures is too high. We plan to employ efficient access methods based on indices for the computation of the similarity values.

We also plan to develop an active database component to the multimedia database. Such a component will allow us to monitor the evolution of the database over time. In such a component, one would specify certain triggers which would automatically initiate some actions whenever certain prespecified conditions are satisfied. For example, such components permit us to monitor, in real-time, pictures being transmitted from a satellite. In a military application, we can define a trigger that alerts the headquarters whenever there is a significant change in the enemy formation in a short time.

References

- [AHU74] A. V. Aho, J. E. Hopcroft and J. D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley Publishing Company, 1974.
- [Amd93] Amdor, F.G. et al., *Electronic How Things Work Articles: Two Early Prototypes*, IEEE TKDE, 5(4), Aug. 1993, pp611-618.
- [ATY95] A. Aslandogan, C. Thier, C. T. Yu, et al "Implementation and Evaluation of SCORE(A System for COnent based REtrieval of Pictures)", IEEE Data Engineering Conference, March 1995.
- [BPJ93] Bach J. R., Paul S., Jain R., *A Visual Information Management System for the interactive Retrieval of Faces*, IEEE TKDE 93.
- [Car93] Cardenas, A.F. et al.: The Knowledge-based Object-Oriented PICQUERY+ Language, IEEE TKDE, 5(4), Aug. 1993, pp 644-657.
- [CB85] T.R. Crimmins and W.M. Brown, Image algebra and automatic shape recognition, *IEEE Transactions on Aerospace and Electronic Systems*, vol. 21, pp. 60-69, 1985.
- [CCT94] S.K. Chang, G. Costaliola and M. Tucci, *Representing and Retrieving Symbolic Pictures by Spatial Relations*, to appear in Journal of Visual Language and Computing.
- [ChH92] Chang, S.K., and Hsu,A., *Image Information Systems: Where Do We Go from Here?*, IEEE Transactions on Knowledge and Data Engineering, CVol. 4,No. 5, pp 431-442.
- [CHH93] S.K. Chang, T.Y. Hou, and A. Hsu, *Smart Image Design for Large Image Databases*, Large image Databases, 1993.
- [CK81] Chang S K, Kunii T L: Pictorial Data-Base Systems, IEEE Computer, Nov 1981, pp 13-21.
- [CSY84] Chang, S. K., Shi, Q.Y. and Yan, C.W. *Iconic Indexing by 2-D Strings*, IEEE Transactions on Pattern Analysis and Machine Intelligence, July 1984.
- [Che76] Chen P P: *The Entity-Relationship Model Toward a Unified View of Data*, ACM Transactions on Database Systems 1(1), March 1976, pp 9-36.

- [Ch92] Chu, W. et al, *A Temporal Evolutionary Object-Oriented Model and its Query Languages for Medical Image Management*, VLDB 92.
- [DDI95] Day Y. F., Dagtas S., Lino M., Khokhar A., Ghafoor A., *Object Oriented Conceptual Modeling of Video Data*, IEEE Data Engineering 1995.
- [DG94] Dimitrova N., Golshani F., *RX for Semantic Video Database Retrieval*, ACM Multimedia 1994.
- [DRB89] J. S. Deogun, V. V. Raghavan, and S. K. Bhatia, *A theoretical basis for the automatic extraction of relationships from the expert-provided data*, Proc. of the Fourth International Symposium on Methodologies for Intelligent Systems: Poster Session Program, pages 123-131, Oak Ridge National Lab., Charlotte, NC, October 1989.
- [Eg89] Egenhofer M. J., *A Formal Definition of Binary Topological Relationships*, in W. Litwin and H. J. Schek, editors, Third International Conference on Foundations of Data Organization and Algorithms (FODO), Paris, France, Lecture Notes in Computer Science, Vol 409, pages 271-286, Springer-Verlag, New York, NY, 1989.
- [GrM92] Grosky W. and Mehrotra, R., *Image database Management*, Advanced in Computer, Vol. 34, Academic Press, New York, pp. 237-291.
- [GrM90] Grosky W. and Mehrotra, R., *Index-Based Object Recognition in Pictorial Data Management*, Computer Vision, Graphics, and Image Processing, No. 52, pp 416-436.
- [GR94] V. N. Gudivada and V. V. Raghavan, *Design and evaluation of algorithms for image retrieval by spatial similarity*, To appear in ACM Trans. on Information Systems, 1994.
- [GRS93] V. N. Gudivada, V. V. Raghavan, and G. S. Seetharaman, *Identification and analysis of semantic attributes in face image databases*, Third symposium on Document Analysis and Information Retrieval, accepted, April 1994.
- [GRV94] Venkat N. Gudivada, Vijay V. Raghavan, and Kanonkluk Vanapipat *A Unified Approach to Data Modeling for a Class of Image Database Applications* Tech. Report 1994
- [GWJ91] Gupta, A., Weymouth, T., and Jain, R. *Semantic Queries with Pictures: The VIMSYS Model* International Conference on Very Large Data Bases, Barcelona, Spain, pp.69-79 1991
- [GZCS94] Gong Y. et al, *An Image Database System with Content Capturing and Fast Image Indexing Abilities* IEEE Multimedia Conference, 1994.
- [HOP91] S.A. Hawamdeh, B. Ooi, R. Price, T. Tang, Y. Deng and L. Hui, *Nearest neighbor searching in a Picture Archive System*, International Conference on Multimedia Information Systems, 1991, edited by Christodoulakis and Narasimhalu.
- [LeeW93] Lee, Eric and Thom Whalen, *Computer Image Retrieval by Features: Suspect Identification*, Proceedings ACM Conferences on Human Factors in Computing Systems, Amsterdam, April 1993. pp. 494-498.
- [LSY89] S. Y. Lee, M. K. Shan, and W. P. Yang, *Similarity retrieval of ICONIC image databases*, *Pattern Recognition*, 22:675-682, 1989.

- [Lee88] Lee, Edward T, *Relationship Hierarchy for picture Representation Using Entity Relationship Diagrams*, *Kybernets*, 17(3), 1988, pp 45-51.
- [LW] Lum V., Wong K., *A Model and Technique for Approximate match of Natural Languages Query*, Korea Info Science Conference 1993.
- [LY93] Liu C., Yu C., *Performance Issues in Distributed Query Processing*, *IEEE Transactions on Parallel and Distributed Systems* 1993.
- [MP92] Manna Z., Pnueli A., *The Temporal logic of Reactive and Concurrent Systems*, Springer-Verlag 1991.
- [MS93b] Marcus S., Subrahmanian V.S., *Structured Multimedia Database Systems*, Technical Report, University of Maryland, 1993.
- [Ni93] Niblack W. et al, *The QBIC Project: Querying Images by Content Using Color, Texture and Shape*, IBM Research Report, Feb 1993.
- [NT89] Naqvi S., Tsur S., *A Logical Language for Data and Knowledge Base*, Computer-Science Press, 1989.
- [OS95] Ogle V., Stonebreaker M., *Chabot: A System for Retrieval from a Relational Database of Images*, Technical Report, UC Berkeley 95.
- [RM89] Reiter R., Mackworth A. K., *A Logical Framework for Depiction and Image Interpretation*, *Artificial Intelligence* 41, 1989.
- [RP92] Rabitti, F and P Savino, *An Information Retrieval Approach for Image Database*, VLDB, Canada, August 1992, pp 574-584.
- [RS91] Rabitti and P Savino, *Image Query Processing Based on Multi-level Signatures*, ACM/SIGIR, USA, October, 1991, pp 305-314.
- [R92] Ramakrishnan R., Srivasta D., Sudarshan S., *Coral-Control, Relations and Logic*, VLDB92.
- [Salt89] Salton G. : "Automatic Text Processing", Addison Wesley, Mass., 1989.
- [SW95a] Sistla A. P., Wolfson O., *Temporal Trigger in Active Database Systems*, *IEEE TKDE* Vol. 7 No. 3, June 1995.
- [SW95b] Sistla A. P., Wolfson O., *Temporal Conditions and Integrity Constraints in Active Database Systems*, Proceedings of ACM SIGMOD 1995, Sanjose, California.
- [SY95] Sistla A. Prasad, Yu Clement, *Retrieval of Pictures Using Approximate Matching*, To appear in *Multimedia Databases*, Springer-Verlag 95.
- [SYH94] A. Prasad Sistla, Clement Yu, R. Haddad, *Reasoning About Spatial Relationships in Picture Retrieval Systems*, VLDB 94.
- [SYLL95] A. Prasad Sistla, Clement Yu, C. Liu, K. Liu, *Similarity Based Retrieval of Pictures Using Indices on Spatial Relationships*, VLDB 95.
- [TaY84] Tamura, H. and Yokoya, N., *Image Database System: A Survey* *Pattern Recognition*, Vol. 17, No. 1, pp. 29-43.

- [TCC91] M. Tucci, G. Costagliola and S.K. Chang, *A Remark on NP-completeness of Picture Matching*, Information Processing Letters, 1991.
- [TS91] Tomisha, Kamada and Satoru, Kawai, *General Framework for Visualizing Abstract Objects and Relations*, ACM Transactions on Graphics, 10(1), 1991, pp 1-39.
- [TZ86] Tsur S., Zaniolo C., *LDL: A Logic Based Data Language*, VLDB, 1986.
- [WAL93] Wu J., Ang P., Lam P., Moorthy A., Narasimhalu A., *Facial Image Retrieval, Identification and Inference System*, Proceeding of ACM Multimedia 93.
- [YG94] Yan T. W., Garcia-Molina H., *Index Structures for Selective Dissemination of Information under the Boolean Model*, ACM TODS 1994.
- [YWLY95] Yang Q., Wu J., Liu C., Yu C., *Efficient Processing of Fuzzy SQL Queries*, IEEE Data Engineering Conference 1995.

Multimedia Information Systems

Bob Allen
rba@bellcore.com

Abstract

While extensive standards have been developed for the representation of objects within complex multimedia documents, much less attention has been paid to the usability of these documents. Key features of electronic text-document browsers are identified and the ways these might be applied to multimedia documents are discussed. Some of these features include structured views, linking, and indexing.

One set of interfaces for digitized multimedia lectures demonstrate how tables of contents can provide access to high-level multimedia structures. A second set of structured interfaces are based on timelines, and they provide an effective aid for understanding relationships among events. Historical information could be displayed to a user and the user could browse for additional information as needed. Finally, two types of digital library interfaces will be described for navigating and searching large collections of documents. One of these interfaces is based on book records organized by the Dewey Decimal Classification. The other interface provides access to records describing computer science documents classified by the ACM Computing Reviews system. These Digital Library interfaces could also be used to structure collections of World-Wide Web pages.

Managing End-System Resources for Predictable Quality-Of-Service

Raj Yavatkar
Department of Computer Science
University of Kentucky
Lexington, KY 40506-0046
raj@dcs.uky.edu

July 27, 1995

1 Introduction

Rapid and explosive advances in network bandwidth rates and processor speeds are enabling new applications based on high-speed communication and distributed processing. Examples of such applications include medical imaging, remote visualization, on-demand video servers, data fusion, distance learning, and virtual reality that underlie some of the Grand Challenge Problems identified under the HPCC (High Performance Computing and Communication) initiative.

These applications typically run on high-performance workstations equipped with devices such as high-resolution bitmap displays, cameras, microphones, and speakers. They make extraordinary demands on underlying networks and computing resources. In particular, these applications expect certain quality-of-service (QOS) guarantees from the underlying system. For instance, an on-demand video player must retrieve video frames from a remote server and play them back at 30 frames per second without any noticeable frame jitter.

Providing such guarantees poses interesting research problems in the areas of network protocols and operating systems. A considerable amount of work is in progress in the area of network protocols. The previous work mainly concentrates on the problems of bandwidth management and switch-based scheduling to provide deterministic or statistical guarantees on end-to-end delay, throughput, and packet losses. The solutions proposed in this area are valuable in ensuring certain QOS for traffic travelling from one end of the network to another for a pair of communicating hosts. However, the other aspect of QOS management, namely, "the

problem of end-system QOS management” arises because it is NOT only sufficient to ensure that the network traffic is delivered with desired QOS across a path through the network, but it is also essential to supplement the network QOS with mechanisms that ensure that data can be delivered (and processed) in a timely fashion across the data path inside the end-system. The data (and control) path inside an end-system connects the network interface with the source (or sink) of network communication such as a multimedia application running in the user space. To guarantee end-system QOS support, one must consider contention for resources such as network interface, CPU (time), memory, and bus bandwidth.

Compared to the research in the area of network-level QOS guarantees, relatively little work has been done in the area of end-system architectures for QOS management. Current, general-purpose operating systems such as Unix do not include support for QOS specification or policies and mechanisms needed to provide predictable service to individual applications.

2 Our Approach

We are investigating research issues in managing end-system resources in an integrated fashion so that an end-system can deliver desired QOS to multimedia applications. Our research is in the context of a new resource management architecture (called AQUA for Adaptive End-system Quality of Service Architecture) that includes a common framework for managing resources such as CPU, network interface, memory, and bus bandwidth.

AQUA consists of the following components:

- QOS Specification and Mapping: a common language between applications and the OS/network subsystem so that an application can specify its QOS requirements and the system can map a specification into corresponding resource requirements.
- Resource Managers: resource reservation and admission control algorithms to set aside resources and to avoid over-commitment of resources.
- Application-level Adaptation: an application-level QOS manager that can change the application’s behavior (e.g. playback rate or display resolution) to adapt to changes in demand and availability of resources.

- Advance Notification: For playback applications, the component keeps track of resource requirements and notifies the OS and network in advance about the impending changes in resource requirements for dynamic channel management and QoS control.

AQUA design is based on the following assumptions:

- A1.** Multimedia applications (MM applications) are not hard real-time applications and can easily tolerate occasional delays in execution from one execution to another. However, MM applications do expect and require a steady rate of progress; for instance, an MPEG player must be able to execute and play back frames at a certain rate. The acceptable rate of progress is typically specified as a range of values and occasional variations within the range are acceptable.
- A2.** The amount of resources needed per execution is not typically known in advance for MM applications. For instance, the amount of data transfer and computing time needed for a MPEG stream varies from one frame to another and overall resource requirement can change significantly when a change in scene occurs.
- A3.** Many CBR and VBR applications can gracefully adapt to resource overloads by reducing their individual resource requirements. Examples of such adaptations include reducing spatial or temporal resolution of video and adaptive audio playout.

2.1 Application-based Graceful Adaptation

A significant idea in AQUA is that it does not require applications to specify all of their resource requirements in advance. Instead, each resource manager makes provisions to adapt to changing resource requirements of admitted applications. The adaptation component in AQUA is based on the principle of Phased Locked Loops (PLL) applied separately to each managed resource.

The PLL consists of a resource manager, a low-pass filter, and an application-level QoS manager (provided as a library routine). As an application executes and uses a resource, the resource manager monitors its rate of progress and provides a feedback to the application manager that indicates the measured QoS. The feedback is provided using a low-pass filter to avoid reacting to transient changes in measured QoS. The

application-level manager compares the measured QOS against the desired QOS (using a comparator) and takes appropriate actions to reduce its resource requirement based on an application-specific policy.

Another component of the adaptation mechanism involves reacting to availability of additional resource as well as to resource overloads. An increase in resource availability should allow existing processes to receive better QOS or an overload might require all admitted processes to reduce their resource requirements. In the case of resource overload, for instance, the resource manager can compute desired reduction in each process's share of its capacity. However, the resource manager cannot assume any knowledge of the application-level policies for reacting to such a resource overload. Therefore, the resource manager simply passes on the request for adaptation to the application-level manager and the manager then adjusts the application's execution rate to reduce its share of the resource capacity.

2.2 Advanced Notification

For VBR applications involving playback, significant changes in resource requirements can be predicted in advance. Resource requirements for a MPEG stream, for instance, change proportional to the size of frame sizes and the frame sizes show periods of gradual increase or decrease. Such changes are handled easily by the PLL in AQUA. However, the resource requirements often go through various phases when frame sizes show sudden and significant jumps as a result of scene changes.

Such phase changes can cause significant performance degradation until the PLL can adjust to them. To avoid such fluctuations, we can store the information on various phases and their corresponding resource requirements with an MPEG stream. A signaling channel could carry this information in advance allowing the network and the end-systems to adapt to the impending changes in resource requirements before the actual change happens. Thus, when the data rate increases the system would have already adapted to the change. This information can also be used by the network to provide dynamic channel management and QoS control.

The advance notification component of AQUA will use a network-layer channel signaling mechanism to convey such information to communicating parties. We propose to further investigate the possibility of using such notification in dynamic resource reservation across an ATM network.

3 Current Status

Currently, we are implementing the proposed framework on Unix workstations interconnected with an ATM LAN.

So far, we have applied the proposed framework to the problem of CPU scheduling by designing a new CPU management algorithm called Rate-based Adjustable Priority Scheduling (RAP). RAP makes the following new contributions. First, it does not assume a priori knowledge of compute times for the MM applications and thus the admission control algorithm estimates and allocates available capacity to new processes. Second, RAP monitors the average computing time needed by a process and its admission control algorithm only guarantees average rates of execution based on average compute times. Third, priorities of individual processes can change in response to changes in execution rates to adapt to varying compute times. Fourth, the admission control algorithm includes conditions for graceful reduction in compute times or rates of execution.

We have evaluated RAP for VBR applications using traces of execution times needed by the Berkeley MPEG player on different MPEG streams and obtained impressive results. In particular, our results show that the Aqua framework can successfully control the rate jitter and provide a steady rate of progress for competing applications in the presence of dynamic changes in resource requirements and overload conditions.

We continue to evaluate the AQUA framework further and plan to extend it to the management of resources such as network interface adaptor, bus bandwidth, and memory (network buffers).

A Framework for Conceptualizing Structured Video

Martha L. Escobar-Molano and Shahram Ghandeharizadeh

Computer Science Department
University of Southern California
Los Angeles, California 90089

September 11, 1995

1 Introduction

Video has been available in a variety of formats since the late 1800s: In the 1870s Eadweard Muybridge created a series of photographs to display a horse in motion [Enc92]. Thomas Edison patented a motion picture camera in 1887 [Enc92]. Needless to say, video has enjoyed more than a century of research and development to evolve to its present format. During the 1980s, digital video started to become of interest to computer scientists. Repositories containing digital video clips started to emerge. This has resulted in a growing interest for a system that supports queries that manipulate and retrieve digital video based on their content.

There are two alternative approaches to represent video: stream-based and structured [Gha95]. With the stream-based approach, a video clip is represented as a sequence of frames that must be displayed at a certain rate (e.g., 30 frames per second). With the structured approach [EM94], a video clip is represented as a collection of objects (e.g., a 3D representation of Simba¹, a 3D representation of a tree) with spatial and temporal constraints (e.g., positions of Simba and a tree in the scene, and the time of their appearances) along with their rendering features (e.g., light intensity, view point). Presently, the structured approach is used to produce animated sequences, e.g., an animated children's show, named "Reboot" [Ber94].

Structured presentations provide for both re-usability of information and development of effective query processing techniques. They enable a user to extract a character (e.g., Simba), a motion path (e.g., the trajectory and timing of Simba's motion) from one presentation and re-use it in another. Moreover, the temporal and spatial information of a structured video can be queried. To illustrate, consider the animation "The Lion King." Assuming it was represented using the structured approach, to retrieve the scene where Simba finds his father (Mufasa) dead, a user can

¹A character in the Disney movie The Lion King.

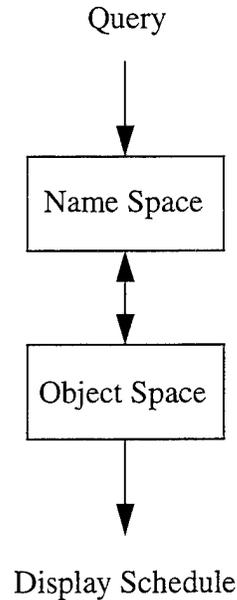


Figure 1: Conceptual Model.

pose the following query: select scenes that contain both Simba and his father such that Mufasa is static while Simba is moving. This enables the user to retrieve the relevant data by issuing a query instead of browsing the video object.

With the current authoring tools, the user has to browse through the file system to select the relevant files to be considered in the animated sequence. This approach is appropriate for a system with few objects. However, it becomes cumbersome as the number of objects and presentations increase (the number of files increases). Moreover, it is not suitable for querying.

This study focuses on the data organization for efficient retrieval of structured video. It proposes a conceptual model (Figure 1) of structured video that consists of an *object space* and a *name space*. The object space represents the rendering aspect of video, e.g., a collection of 3D representations of characters and backgrounds associated using temporal and spatial constraints along with rendering characteristics. The name space represents the user interpretation of video, e.g., name of characters, actions such as running, walking, etc. When a user poses a query against the name space, the system searches the name space and possibly the object space to obtain the information required. If the output of the query is a video, then it also delivers the video specification to the rendering package according to the temporal constraints. For example, suppose that the name space of a movie contains the names of the backgrounds, and descriptions of the motions in the scenes. Each description consists of the name of the character and the type of motion (e.g., running, walking, etc.). To process the query “show me the scenes with Simba running in the jungle,” the system selects the scenes with a jungle as their background and Simba running as their

motion from the name space. Then, it obtains a schedule of objects in the selected scenes with their times of appearance (termed *display schedule*) from the object space. The display schedule is then processed to generate a retrieval schedule [EMGI95, EMG95] that satisfies the temporal constraints. Based on the retrieval schedule, the system delivers data from disk to the rendering package to display the results. We focus on the conceptual model for the name and object spaces.

We first present a layered representation of the object space in Section 2. Next, we introduce the name space as an intermediary between the user and the object space in Section 3. And finally, we conclude with future directions of this research. We do not describe a model for translating a stream-based presentation to a structured one. However, this research direction is an important one in itself.

Temporal representation and its manipulation were studied prior to the advent of multimedia. Allen [All83] introduced an interval-based representation of time and a reasoning algorithm to maintain it. With the introduction of continuous media types such as video, researchers have studied temporal modeling in the context of databases. Oomoto and Tanaka [OT93] proposed an object-based data model with interval-inclusion based inheritance. Gibbs et al [GBT94] proposed a timed stream as the basic element of time-based media. Their study introduced a basic structuring mechanisms for timed streams. Little and Ghafoor [LG93] presented an interval-based representation of temporal relationships between multimedia streams as part of a database. Their study introduced algorithms for forward, reverse and partial playout of the streams based on the introduced representation. Schloss and Wynblatt [SW94] presented a conceptual object-oriented database to represent multimedia data. It divides the multimedia database in layers for modularity and re-usability purposes. It considers a multimedia database as a collection of streams (e.g., audio, video) with temporal relationships, and synchronization policies (i.e., strictness and resynchronization after failure). In sum, these studies organized the frames of a stream-based representation and did not consider the spatial and temporal relationship between the objects that participate in the individual frames of a video. Moreover, time intervals were used mainly for modeling purposes and not for querying purposes.

Spatial and temporal relationships of objects that constitute a multimedia document were investigated by Weitzman and Wittenburg [WW94]. Their study defines a presentation as a collection of objects and relations (e.g., author of, title of), a relational grammar, and a collection of constraints (spatial, temporal, graphics attributes). This abstraction has similarities with our abstraction of structured video: objects correspond to atomic objects; relational grammars to component-composed object relations; and constraints to spatial and temporal associations, and rendering features. However, their study does not address the issues of access and filtering of information.

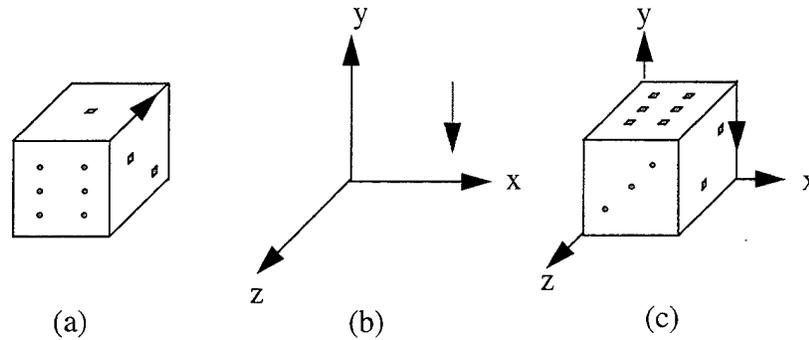


Figure 2: Spatial Constructs: (a) object o and vector v_o . (b) rendering space and vector v_r . (c) the placement of o in the rendering space after applying the spatial construct: $v_o \rightarrow v_r$.

2 Object Space

The object space of structured video consists of a collection of objects, spatial and temporal constructs, and rendering features. The spatial and temporal constructs define where in the rendering space and when in the temporal space the component objects are displayed. The rendering features define how the objects are displayed.

A *Rendering Space* is a coordinate system defined by n orthogonal vectors, where n is the number of dimensions (i.e., $n = 3$ for 3D, $n = 2$ for 2D). A spatial construct specifies the placement of a component in the rendering space. For example, the representation of a dining room (4 walls, a table and 8 chairs) consists of a 3D coordinate system (rendering space) and 13 spatial constructs, one for each object. These spatial constructs define unambiguously the location of the walls, the table and the chairs in the rendering space. They also implicitly define spatial relationships such as the chairs are around the table.

Formally, a *Spatial Construct* of an object o maps a vector v_o in o into a vector v_r in the rendering space. The *placement* of object o in the rendering space is the rigid translation of o into the rendering space so that v_o and v_r coincide (Figure 2).

Analogously, different components are rendered within a time interval, termed *Temporal Space*. A temporal construct specifies the subinterval in the temporal space when the component object is rendered. For example, consider a presentation that consists of five scenes of one minute each. Suppose that Simba appears in the first three scenes and Mufasa during the last three scenes. The temporal space of the presentation is the time interval $[0, 5]$. The temporal constructs for the five scenes are $[0, 1]$, $[1, 2]$, $[2, 3]$, $[3, 4]$, and $[4, 5]$, respectively. These constructs implicitly define temporal relationships such as Simba and Mufasa appear simultaneously during the third scene.

Motion representations have both temporal and spatial constructs. A motion consists of a path

and a sequence of postures p_1, \dots, p_n that provide the illusion that the object is moving. These postures must be displayed at the appropriate times and follow the path to provide the continuation of the movement. A motion is represented by a collection of postures, each posture may employ a temporal and a spatial construct. The spatial constructs determine the path followed by the postures and the temporal constructs the timing of their appearances. For example, consider the scene of a thief chased by a policeman. It contains two motions, one for each character. Each motion consists of a path, a sequence of postures and the timing of postures appearances. This information implicitly defines relationships such as chasing.

There are three layers to represent the object space:

- (i) *Atomic Objects* that consist of objects participating in the presentation. These objects are indivisible (i.e., they are displayed in their entirety).
- (ii) *Composed Objects* that consist of objects constrained using temporal and spatial constructs.
- (iii) *Rendering Features* that consist of rendering attributes such as camera position, light sources, etc.

Example 2.1: Suppose Mickey Mouse walks along a path in a scene (Figure 3). Then, we have the 3D representations of different postures of Mickey Mouse in the Atomic Objects layer. For example, his posture when he starts walking, his posture one second later, etc. They are denoted by p_1, p_2 , etc. These postures might have been originals composed by an artist or generated using interpolation. We also include the 3D representation of the background (denoted by sc_1) in the Atomic Objects layer.

To represent the motion, we have to specify spatial and temporal constructs in the Composed Objects layer. The spatial constructs for the motion representation are specified by the curve labeled c_1 . The curve illustrates the path followed by Mickey Mouse (i.e., the different positions reached) and the vectors the placement of Mickey Mouse postures in the rendering space. The temporal constructs are specified by the triplets: component, starting time and duration, representing the posture of the character and its timing. For example, the point labeled $(p_1, 0, 1)$ indicates that the posture p_1 is at the scene at the vector associated with the point and appears at time 0 and lasts for 1 second.

To associate the motion of Mickey Mouse with the background, we have the spatial and temporal constructs in the Composed Objects layer represented by c_2 . The spatial constructs define where in the rendering space the motion and the background are placed. The rendering space is represented by the 3D coordinate system and the placement of the motion and background by the vectors. The motion is mapped to the vector labeled $(c_1, 5, 25)$ and the background to the other

<p>RENDERING FEATURES</p> <p>Rendering characterization of composed objects</p>	<p>View point, light sources, etc. of each scene in the movie.</p>
<p>COMPOSED OBJECTS</p> <p>Temporal and spatial association to objects</p>	<p>The diagram shows two 3D coordinate systems. The first, labeled 'c1', has three axes. A wavy line representing a path starts at point (p1, 0, 1), goes to (p2, 1, 2), and then to (p3, 3, 2). The second, labeled 'c2', also has three axes. A point (c1, 5, 25) is marked on one axis, and another point (sc1, 0, 30) is marked on another axis.</p>
<p>ATOMIC OBJECTS</p> <p>Indivisible objects</p>	<p>Different postures of Mickey Mouse: p1, p2, ... A scenery with a house, trees, mountains, and a meadow: sc1.</p>

(a)

(b)

Figure 3: Levels of abstraction in the object space. (a) Description of each level (b) Example of how a movie with a scene where Mickey Mouse walks is represented.

vector. The temporal constructs define the timing of the appearances of the background and the motion. For instance, the background *sc1* appears at the beginning of the scene while Mickey Mouse's motion (*c1*) starts at the 5th second.

Finally, the rendering features are assigned by specifying the view point, the light sources, etc., for the time interval at which the scene is rendered.

2.1 Atomic Objects

Video applications may represent atomic objects in alternative ways (e.g., wire-frame, surface, solid representation, etc.) at the physical level. From a conceptual perspective, these physical representations are considered as an unstructured unit (i.e., a BLOB). Atomic Objects can also be represented at the conceptual level as either:

- (i) A procedure that takes some parameters as inputs and constructs a BLOB that represents the object. For example, a geometric figure can be represented by the parameters of its parts (i.e., the radius, the length of a side of a square, etc.) and a procedure that consumes those parameters and produces a bitmap that represents the object. This type of representation is termed *Parametric*.
- (ii) An interpolation of two other atomic objects. For example in animation, the motion of a character can be represented as postures at selected times and the postures in between can be obtained by interpolation. As in animation, this representation is termed *In-Between*.
- (iii) A transformation applied to another atomic object. For example the representation of a posture of Mickey Mouse can be obtained by applying some transformation to a master representation. This representation is termed *Transform*.

In Figure 4, we present the schema of the type Atomic that describes these alternative representations. The conventions employed in this schema representation as well as others presented in this paper are as follows: The names of built-in types (i.e., strings, integers, etc.) are all in capital letters as opposed to defined types that use lower case letters. ANYTYPE refers to strings, integers, characters and complex data structures. A type is represented by its name surrounded by an oval. The attributes of a type are denoted by arrows with single line tails. The name of the attribute labels the arrow and the type is given at the head of the arrow. Multi-valued attributes are denoted by arrows with two heads and single value attributes by arrows with a single head. The type/subtype relationship is denoted by arrows with double line tails. The type at the tail is the subtype and the type at the head is the supertype.

For example, in Figure 4 *Parametric* is a subtype of *Atomic*, and it has two attributes: *Parameters* and *Generator*. *Parameters* is a set of elements of any type and *Generator* is a function

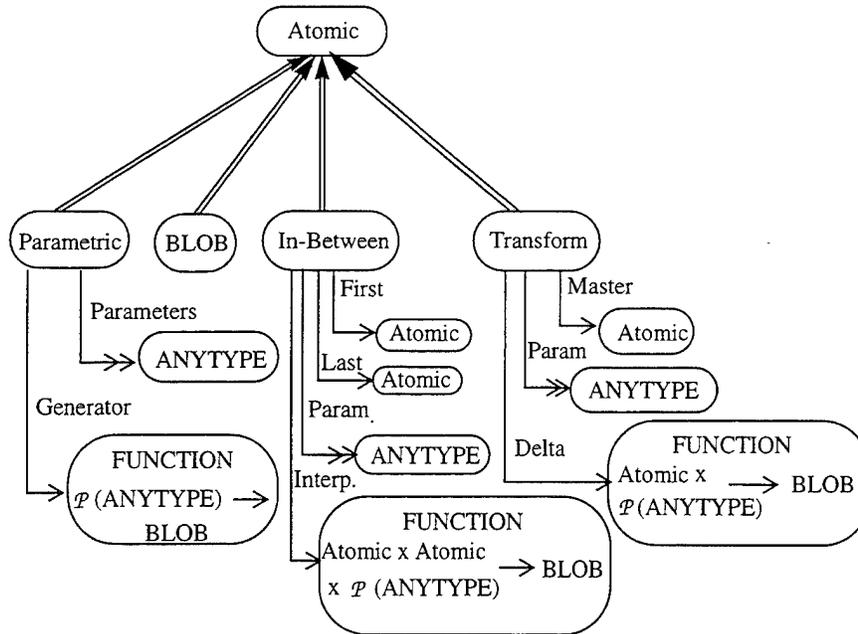


Figure 4: Atomic object schema

that maps a set of elements of any type (i.e., Parameters) into a BLOB.

2.2 Composed Objects

Composed Objects are collections of objects with spatial and/or temporal constructs.

Definition: A *Composed Object* C is represented by the set:

$$\{(e_i, p_i, s_i, d_i) \mid \begin{array}{l} e_i \text{ is a component of } C, \\ p_i \text{ is the mapped vector in } C\text{'s rendering space defined by} \\ \text{the spatial construct on } e_i, \text{ and} \\ [s_i, d_i] \text{ is the subinterval defined by a temporal construct on } e_i \end{array}\}$$

A composed object may have more than one occurrence of the same component. For example, a character may appear and disappear in a scene. Then, the description of the scene includes one 4-tuple for each appearance of the character. Each tuple specifies the character's position in the scene and a subinterval when the character appears.

The definition of composed objects establishes a hierarchy among the different components of an object. This hierarchy can be represented as a tree. Each node in the tree represents an object with spatial and temporal constructs (i.e., the 4-tuple in the composed object representation: $(component, position, starting\ time, duration)$), and each arc represents the relation *component of*.

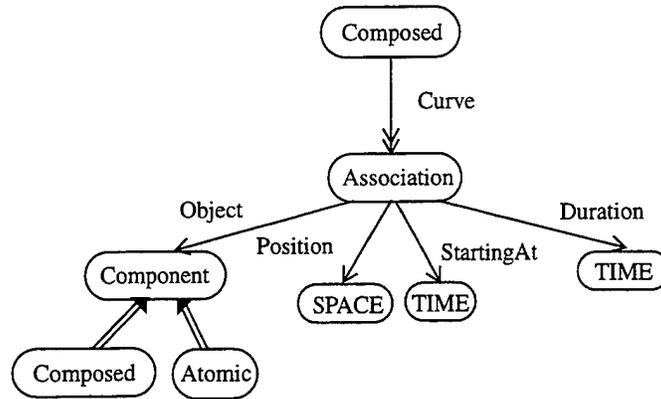


Figure 5: Composed Object schema

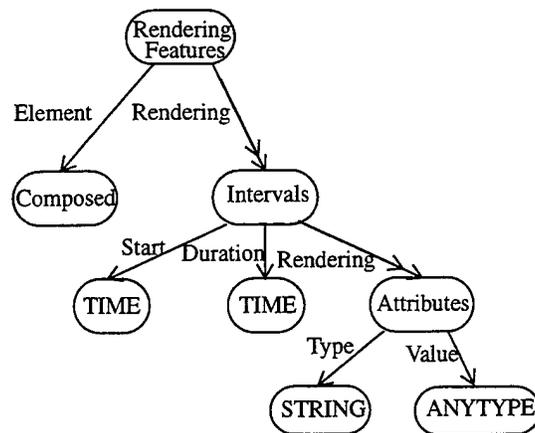


Figure 6: The rendering features schema

Figure 5 illustrates the schema associated to the representation of spatial and temporal constructs in the Composed Objects layer.

2.3 Rendering Features

Rendering features are associated to intervals in the temporal space of a composed object. They are a collection of tuples, $(descriptor, value)$, to represent the description and value of a feature. Figure 6 represents the schema associated to the rendering features layer.

3 Name Space

Having the name space as intermediary between the user and the object space facilitates query processing. The name space consists of descriptive data based on the user's interpretation of the video. Instead of using the rendering data in the object space to pose a query the user employs descriptions (e.g., names, adjectives). For example, the user might ask for a scene with a stop sign in it instead of a scene with an hexagon and the word "STOP" inside.

A mapping from the name space to the object space allows the system to display the sections of the video that correspond to a given description. For example to process the query "show me the scenes where Simba finds Mufasa dead", the system searches the name space for scenes with Mufasa static and Simba moving. Then, it uses the mapping to the object space and display the corresponding scene.

A mapping from the object space to the name space allows to re-use not only the rendering aspect of an object but also its descriptive information. When a user extracts a character from one video to be used in another, he/she should be able to extract the descriptive data as well (e.g., name of the character). Moreover, the information in the object space is useful to obtain descriptive data. For example, the temporal and spatial constructs can be used to compute the type of motions (e.g., running, walking, etc.).

Descriptive information associated to component objects might be useful to describe the composed object. Applying the union of descriptive data associated to components to obtain the description of the composed object might lead to excessive and inadequate description of the object. For example, consider a scene with a policeman chasing a thief. Suppose that the motions of the policeman and the thief are described as "running" when considered separately. If the system applies the union of the descriptive data of the two motions to describe the scene, then the result is a scene with a policeman running and a thief running. However, a more adequate description of the scene would be as having a policeman chasing a thief. Therefore, another aspect of the name space is how to combine descriptive data to obtain the description of composed objects.

The name space consists of: (1) user defined data (e.g., names of the characters), (2) derived data (e.g., types of motions), and (3) a combination strategy (e.g., a scene with character B smiling and laughing can be described as containing B in a happy mood). User defined data is represented as attributes of atomic and composed objects. Derived data is represented as methods of atomic and composed objects that might use the temporal and spatial constructs associated to the object. Combination strategies are defined using rules that specify how to obtain the description of an object from the description of its components.

A rule consist of a condition and an action. The condition is a conjunction of positive atoms

and the action is either “insert an object o in the database” or “set object o to a value v in the database”. Besides defining the requisites to perform the action, the condition constrains object o and value v in the action. For example, suppose that a character appears in a scene smiling, crying, laughing and complaining. The condition of a rule that generalizes the temper of the character to “moody” verifies that the character has these different moods in the same scene. The condition also bounds object o to the temper of the character, so that the action of the rule “set o to moody” would update the proper object.

We now define the syntax of the rules. The literals in the condition of a rule are predicates on terms of the following forms:

- (i) x : x is a variable.
- (ii) c : c is a constant.
- (iii) $[att_1 : t_1, \dots, att_n : t_n]$: t_1, \dots, t_n are terms, and att_1, \dots, att_n are attribute names.
- (iv) $x.att_1.att_2 \dots att_n$: x is a variable, att_1 is an attribute of object with oid x , att_2 is an attribute of object with oid $x.att_1$, and so forth.
- (v) $x.att_1.att_2 \dots att_m(t_1, \dots, t_n)$: if x is a variable, t_1, \dots, t_n are terms, att_1 is an attribute of object with oid x , att_2 is an attribute of object with oid $x.att_1$, and so forth. att_m is a method that applies to an object with oid $x.att_1.att_2 \dots att_{m-1}$.

The condition of a rule is a conjunction of predicates of the following forms:

- (i) $t \in C$: t is a term and C is a class name.
- (ii) $t \in x$: t is a term and x is a variable whose value is a set.
- (iii) $t = x$: t is a term and x is a variable.

The action of a rule is one of the following:

- (i) Insert t in C : t is a term and C is a class name.
- (ii) Insert t in x : t is a term and x is a variable whose value is a set.
- (iii) Set x to t : t is a term and x is a variable.

The semantics of the rule is as follows: Given a variable assignment, if all atoms in the condition are true, then perform the action unless there is a constraint that is violated if the action is performed. For example, consider the rule with $x \in A, x \in B$ as its condition and “Insert x in

C as its action. Suppose that A , B and C are class names. The meaning of the rule is: for any variable assignment, if the value assigned to x is an object of type A and B in the database, then the value assigned to x is included in C , unless the value is already in C or there is a constraint that impedes the inclusion.

We constrain the rules so that the variable assignments that make the condition true are limited to: constants, values in the database, results of finite number of methods calls, or a combination of the above. The rules must satisfy the following: There is an order l_1, \dots, l_n of the literals in the condition such that all variables in the rule are bounded with respect to that order. A variable x is *bounded* with respect to l_1, \dots, l_n if and only if there is a literal l_i in the condition that has one of these forms:

- (i) $x \in C$, where C is a class name.
- (ii) $x \in y$, where y is bounded w.r.t. l_1, \dots, l_{i-1} .
- (iii) $x = y$, where y is bounded w.r.t. l_1, \dots, l_{i-1} .
- (iv) $x = c$, where c is a constant.
- (v) $x = [att_1 : a_1, \dots, att_m : a_m]$, where a_1, \dots, a_m are either constants or bound variables w.r.t. l_1, \dots, l_{i-1} .
- (vi) $x = y.att_1 \dots att_m$, where y is bounded w.r.t. l_1, \dots, l_{i-1} .
- (vii) $x = y.att_1 \dots att_m(a_1, \dots, a_k)$; where y is bounded w.r.t. l_1, \dots, l_{i-1} , and a_1, \dots, a_m are either constants or bound variables w.r.t. l_1, \dots, l_{i-1} .

Without loss of generality, we assume that the mappings from the object to the name space and vice versa are defined as two relations, *CombAtomic* and *CombComposed*. These relations have two attributes, *Rendering* and *Description*, that correspond to the rendering information and description of an object, respectively.

To illustrate the previous concepts, consider a description for The Lion King that partitions a movie into scenes and each scene into backgrounds and motions. Moreover, each motion has only atomic objects as its components. This description includes the names of the characters and the type of the motions in the video. A name space for this description associates an attribute *Type* to a composed object to identify a scene, a motion and a background. It associates attributes *Name* and *Motions* to atomic and composed objects, respectively. Where *Name* is a string (the character name) and *Motions* is a set of tuples (one for each character). Each tuple in *Motions* has two attributes, *Name* and *MType*, that represent the character name and the type of motion,

Literals	Comments
Rule R1	
Condition	
$m \in CombComposed,$ $m.Description.Type = Motion,$ $o \in CombAtomic,$ $o.Rendering = or,$ $m.Rendering.Curve = path,$ $point \in path,$ $point.Object = or,$ $name = o.Description.Name,$ $motions = m.Description.Motions$	m is a composed object m is a motion o is an atomic object or is the rendering part of o $path$ is the trajectory of the motion $point$ is one component of the motion o is the object in $point$ $name$ is o 's name $motions$ is m 's motions
Action	
Insert $\{Name : name, MType : ComputeMType()\}$ in $motions$	
Rule R2	
Condition	
$s \in CombComposed,$ $s.Description.Type = Scene,$ $m \in CombComposed,$ $m.Description.Type = Motion,$ $m.Rendering = mr,$ $s.Rendering.Curve = path,$ $point \in path,$ $point.Object = mr,$ $motions = m.Description.Motions,$ $motion \in motions,$ $motype = motion.MType(m, motion.Name),$ $scmotions = s.Description.Motions$	s is a composed object s is a scene m is a composed object m is a motion mr is the rendering part of m $path$ are the temporal and spatial constructs of s $point$ is one component of the scene m is a component of the scene $motions$ is m 's motions $motion$ is a motion in m $motype$ is m 's type of motion $scmotions$ identifies s 's motions
Action	
Insert $\{Name : motion.Name, MType : motype\}$ in $scmotions$	

Figure 7: Rules to create motions descriptions for scenes and motions.

respectively. There must be at most one motion per character associated to a composed object. The system would reject an insertion of a motion for a character already in *Motions*.

Suppose that the user assigns the names of the characters to the atomic objects, the method `ComputeMType()` computes the types of motions based on the spatial and temporal constructs, and there is a pre-defined generalization (combination strategy) of motion types in a scene. Then, the character name of a motion can be obtained from the names associated to the atomic objects (Rule R1 in Figure 7). The motion type of a character at the motion level is computed by `ComputeMType()`. And, the motion type of a character at the scene level is defined by a rule (Rule R2 in Figure 7 and R3 in Figure 8).

The effect of rule R1 (Figure 7) is to extract the characters names from the atomic objects and include them in the description of the characters motions. R1 selects a pair $\langle m, o \rangle$ of a motion and one of its atomic components. Then, it extracts the character name N from o and constructs a motion M consisting of N and the method `ComputeMType()`. Then, it includes M in the motions associated with m .

The effect of rule R2 (Figure 7) is to extract a motion description, compute its motion type and include it in the scene description. R2 selects a pair $\langle s, m \rangle$ of a scene and one of its motions. Then, it extracts a motion description MD from m and call the method to compute the motion type of MD . It then constructs a motion description MD' consisting of the name in MD and the computed motion type. Finally, it includes MD' in the motions associated with s .

The effect of rule R3 (Figure 8) and a symmetric rule (as R3 except that the last two literals in the condition are $mmotype = \text{jogging}$ and $scmotype = \text{running}$) is to generalize the motion types `running` and `jogging` into `rapid motion`. R3 selects a pair $\langle s, m \rangle$ of a scene and one of its motions. If s and m have a motion with the same character name and the motion type of the character in s is `running` while the motion type in m is `jogging`. Then, it changes the motion type of the character in s to `rapid motion`.

4 Conclusions and Future Research

In this paper, we presented a conceptual model for structured video (i.e., a collection of objects with spatial and temporal constrains along with their rendering features). The model is partitioned into an object and a name space. The object space corresponds to the rendering aspect of the video, while the name space to the description of the video to facilitate querying processing.

The name space provides mechanisms that facilitates the content-based retrieval. It enable us to employ the rendering information in the object space to describe the video, and to decrease the effort of supplying descriptions of video. Methods can access the temporal and spatial information

Literals	Comments
Rule R3	
Condition	
$s \in CombComposed,$ $s.Description.Type = Scene,$ $m \in CombComposed,$ $m.Description.Type = Motion,$ $m.Rendering = mr,$ $s.Rendering.Curve = path,$ $point \in path,$ $point.Object = mr,$ $mmotions = m.Description.Motions,$ $mmotion \in mmotions,$ $mmotype = mmotion.MType(m, mmotion.Name),$ $scmotions = s.Description.Motions,$ $scmotion \in scmotions,$ $scmotype = scmotion.MType,$ $mmotion.Name = scmotion.Name,$ $mmotype = running,$ $scmotype = jogging,$	s is a composed object s is a scene m is a composed object m is a motion mr is the rendering part of m $path$ are the temporal and spatial constructs of s $point$ is one component of the scene m is a component of the scene $mmotions$ is m 's motions $mmotion$ is a motion in m $mmotype$ is m 's type of motion $scmotions$ identifies s 's motions $scmotion$ is a motion in s $scmotype$ is one of s 's type of motion $mmotion$ and $scmotion$ correspond to the same character $mmotion$ is "running" $scmotion$ is "jogging"
Action	
Set $scmotype$ to rapid motion	

Figure 8: Rule to generalize "running" and "jogging" into "rapid motion"

in the object space to define relations among the objects in the video. These relations can be used as a mechanism to filter the information. Rules define strategies to obtain descriptions from other descriptions therefore increase the descriptive power of the name space.

Future work includes finding the expressive power of the rules, implementing the model, and devising indexing techniques for structured video

References

- [All83] James F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, November 1983.
- [Ber94] S. Bernstein. Techno-artists 'tooning up. *Los Angeles Times*, pages F1,F4, November 10, 1994.
- [EM94] M. L. Escobar-Molano. Management of resources to support continuous display of structured video objects. Technical Report USC-CS-95-616, University of Southern California, 1994.
- [EMG95] M. L. Escobar-Molano and S. Ghandeharizadeh. Continuous display of structured video using a multi-disk architecture. In *Submitted to Extended Database Technology*, 1995.
- [EMGI95] M. L. Escobar-Molano, S. Ghandeharizadeh, and D. Ierardi. An Optimal Resource Scheduler for Continuous Display of Structured Video Objects. *IEEE Transactions on Knowledge and Data Engineering*, 1995.
- [Enc92] *The Software Toolworks Multimedia Encyclopedia*. Software Toolworks Incorporated, 1992.
- [GBT94] S. Gibbs, C. Breiteneder, and D. Tschritzis. Data Modeling of Time-based Media. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, May 1994.
- [Gha95] S. Ghandeharizadeh. Stream-based Versus Structured Video Objects: Issues, Solutions, and Challenges. In S. Jajodia and V.S. Subrahmanian, editors, *Multimedia Database Systems: Issues and Research Directions*. Springer Verlag, 1995.
- [ISK⁺93] H. Ishikawa, F. Suzuki, F. Kozakura, A. Makinouchi, M. Miyagishima, Y. Izumida, M. Aoshima, and Y. Yamane. The model, language, and implementation of an object-oriented multimedia knowledge base management system. *ACM Transactions on Database Systems*, 18(1):1-50, March 1993.
- [LG93] T. D. C. Little and A. Ghafoor. Interval-based conceptual models for time-dependent multimedia data. *IEEE Transactions on Knowledge and Data Engineering*, 5(4), August 1993.
- [OT93] E. Oomoto and K. Tanaka. Ovid: Design and implementation of a video-object database system. *IEEE Transactions on Knowledge and Data Engineering*, 5(4):629-643, August 1993.
- [SW94] G. A. Schloss and M. J. Wynblatt. Building temporal structures in a layered multimedia data model. In *Proceedings of ACM Multimedia*, pages 271-278, October 1994.
- [WW94] L. Weitzman and K. Wittenburg. Automatic presentation of multimedia documents using relational grammars. In *Proceedings of ACM Multimedia*, pages 443-451, October 1994.

Object-Oriented Modeling and Querying of Multimedia Data

(Extended Abstract)

Arif Ghafoor and Young Francis Day
Distributed Multimedia Systems Laboratory
School of Electrical and Computer Engineering
1285 Electrical Engineering Building
Purdue University
West Lafayette, IN 47907-1285

1 Introduction

In this paper, we present an object-oriented approach for modeling multimedia document and propose a query language for accessing documents. The document model is based on our earlier proposed enhanced OCPN model [2] with the additional specification of logical structures. The language utilizes the n -ary relations extensively and provides facilities for declaration of multimedia data, document composition and imprecise retrieval, spatial/temporal composition, and playout controls. Section II outlines the object-oriented model and Section III presents the main features of the language.

2 Object-Oriented Model for Multimedia Data

In order to model spatio-temporal semantics of multimedia data and to formally express composition schema for multimedia document we first discuss the generalized n -ary relations, that we have proposed earlier in [1, 4]. These relations serve as the constructors for data model. The generalized relations are listed in Table 1 and their graphical representations are shown in Figure 1. A generalized n -ary relation $R(\tau^1, \dots, \tau^n)$ is a relation among n

Relation name	Symbol	constraints, $\forall i, 1 \leq i < n$
before	B	$\tau_i^e < \tau_{i+1}^s$
meets	M	$\tau_i^e = \tau_{i+1}^s$
overlaps	O	$\tau_i^s < \tau_{i+1}^s < \tau_i^e < \tau_{i+1}^e$
contains	C	$\tau_i^s < \tau_{i+1}^s < \tau_{i+1}^e < \tau_i^e$
starts	S	$\tau_i^s = \tau_{i+1}^s \wedge \tau_i^e < \tau_{i+1}^e$
completes	CO	$\tau_i^s < \tau_{i+1}^s \wedge \tau_i^e = \tau_{i+1}^e$
equals	E	$\tau_i^s = \tau_{i+1}^s \wedge \tau_i^e = \tau_{i+1}^e$

τ_s^i = starting coordinate of object τ^i , τ_e^i = ending coordinate of object τ^i

Table 1: n -ary relations

intervals, τ^i , $i = 1, \dots, n$ which are located on a single axis with an origin and satisfy one of the conditions in Table 1 with respect to each other.

For document modeling we use a hierarchical approach. At the highest level, multimedia documents can be organized using a Petri-Net-based hypertext model, where each node of the hypertext represents a multimedia document. Users can browse through the Petri Net by following the arrows (links). Each document, in turn, is represented by an enhanced OCPN. Each place in the OCPN represents a multimedia data with duration and spatial display information. The content of the multimedia data is assigned through either direct file assignment or in the form of content-based spatio-temporal event expressions. The logical structure of a multimedia document can be expressed as a sequential connection of *logical units* (places in the OCPN). For compactness, any subnet in OCPN involving parallel temporal relations can be viewed as a logical unit. The logical structure is specified by labeling the transitions in the OCPN.

The enhanced OCPN includes the following attributes for each place, in addition to the duration information; (i) W_i : the display area of the place; (ii) P_i : the priority vector which describes the relative ordering among background/foreground locations of intersecting spaces (windows) as time evolves; (iii) an ordered set of unary operations (e.g., crop, scale) applied to the data associated with this place, and (iv) in case of image/video data, a logical expression describing their spatio-temporal events (contents).

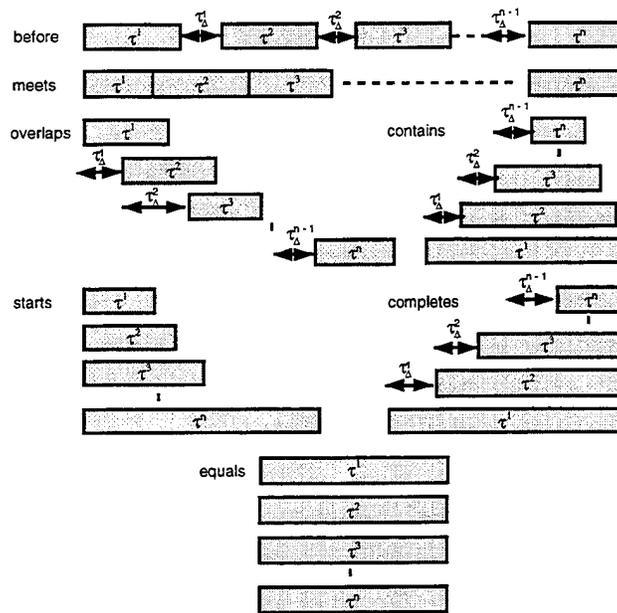


Figure 1: n -ary relations

The enhanced OCPN can be translated into an object-oriented model using the temporal n -ary relations. By specializing these relations in temporal domain, we can view them as classes in an object-oriented environment. The seven n -ary operations can be arranged in a IS-A (or generalization) hierarchy as shown in Figure 2. These classes can be grouped into two super classes, namely, the sequential and parallel temporal operation classes, which correspond to their sequential and parallel nature of operations, respectively. Parallel operations, can be divided based on simultaneous and non-simultaneous starting times. Each temporal operation class has the following attributes: pointers to components, inter-interval delays (if needed), presentation rate, etc. Methods for manipulating these attributes are also defined. The spatial information can be embedded within the temporal objects in the form of the state variables. By using a heuristic algorithm given in [2], an enhanced OCPN can be translated into an hierarchy. The leaf nodes in this hierarchy not only represent the raw data, but also the associated *content*, as mentioned earlier. These contents can be propagated along the path from the leaves to the root (i.e., through the whole document).

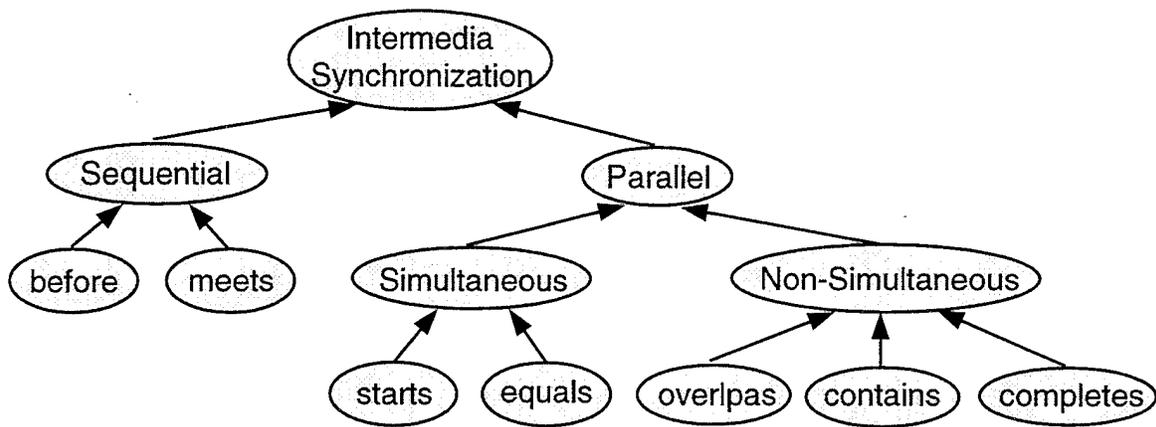


Figure 2: Temporal operation class and its subclasses

The contents can be modeled using n -ary spatio-temporal relations as proposed in [3, 4].

An example multimedia document (for simplicity, we have excluded logical links) with temporal requirement is shown in Figure 3 and with its spatial requirements represented by the enhanced OCPN shown in Figure 4. This OCPN can be translated into an object-oriented abstraction hierarchy (for the temporal aspect) shown in Figure 5. A similar hierarchy can be generated for the logical structure aspect also.

3 A Query Language

Recently we have proposed a query language to retrieve multimedia documents from a database. The language is based on the data modeling approaches mentioned in the previous section. The syntax of the language is based on the predicate calculus [5]. A query may consist of the following four expressions; (i) data declaration declaration for generating an object-oriented database including declaration of object, class, and meta-class, (ii) a composition_expression used for composing/accessing an OCPN-based document, (iii) a retrieve_expression used for content-based retrieval, and (iv) a play_expression for playing back individual multimedia object or a document. These expressions in turn consist of sub-expressions containing object ids, logical operators AND and OR, spatio-temporal operators,

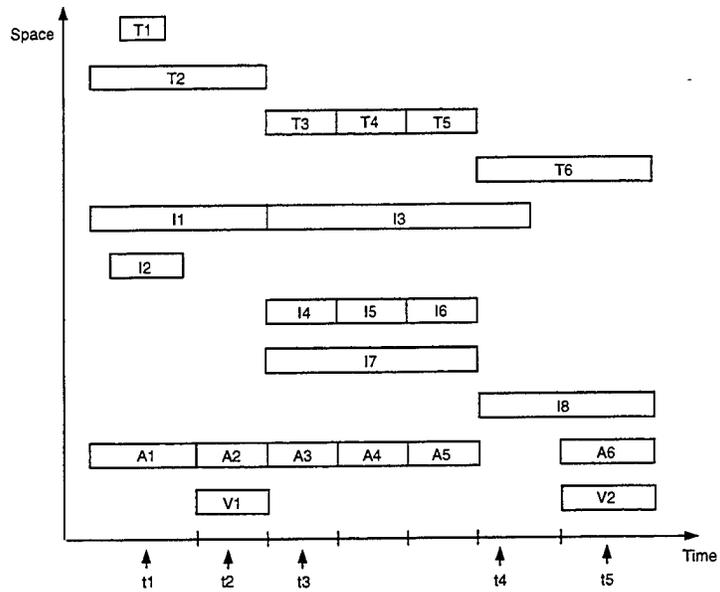


Figure 3: Example timeline

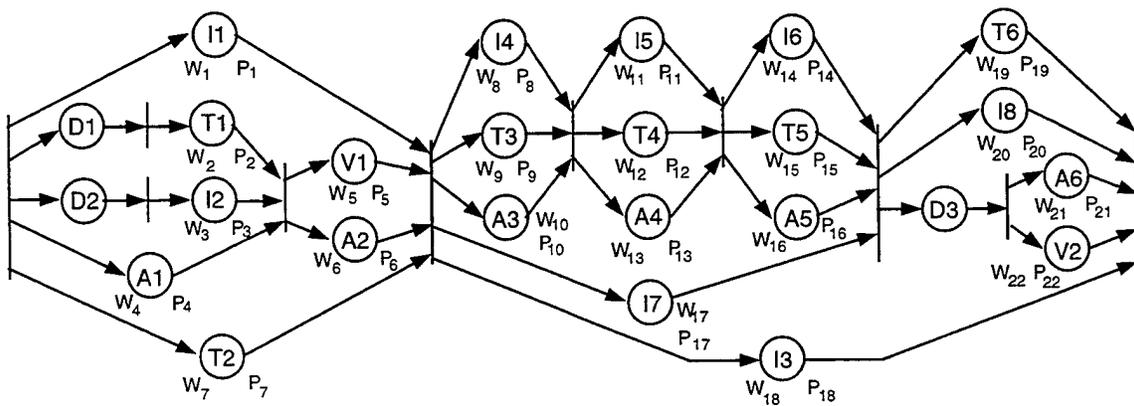


Figure 4: Example OCPN

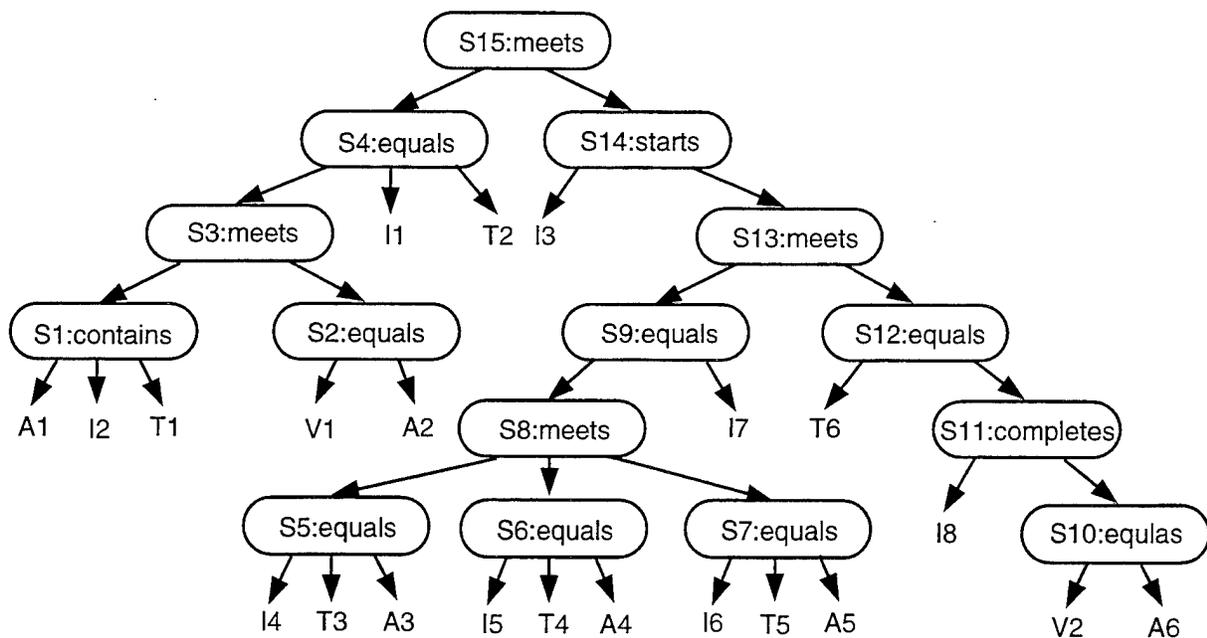


Figure 5: Object hierarchy of example OCPN

and event specification to describe content of image/video object.

Object-Oriented Schema Declaration

Statements in this category are used: to define classes, to create instances of classes, to remove instances/classes, to manipulate raw multimedia data, and to create signatures of raw multimedia data. For example, a class X with superclasses Y and Z , attributes a_1 (domain class C_1) and a_2 (domain class C_2), and methods m_1 can be declared as :

class X subclass-of Y,Z

attributes

C_1 a_1 ;

C_2 a_2 ;

methods

m_1 ;

Another example of the language is applying a sequence of unary spatial operations to a media object. For example, y is an id of an image object x can be formed using assignment

statement $x = y.crop(0, 0, 100, 100).scale(2)$. As a result of this statement, the image y is cropped from upper left corner with 100 pixel by 100 pixel and scaled up with a factor two and the new object is x .

Queries for Multimedia Document Retrieval

In the proposed language, the temporal structure of the document is specified recursively, using the n -ary temporal operators B (*before*), O (*overlaps*), C (*contains*), CO (*completes*), M (*meets*), S (*starts*), or E (*equals*). For displayable objects, we use a *layout* command to specify the associated spatial information.

As an example of using the n -ary operator and *layout* command for constructing multimedia query, consider the OCPN shown in Figure 6 (a), with the timeline shown in Figure 6 (b). The following query specifies the temporal aspect of this document.

$$q_{example_1} := E(M(C(A1, D2 : G1, (D1 - D2) : T2)), E(V1, A2), A3), I1, T1),$$

where $q_{example_1}$ is the surrogate of the document, and $A1$, $D2$, etc, represent variables for the component multimedia data. The composition process is elaborated as follows. $A1$, $T2$, $G1$, $D1$, and $D2$ are replaced by a place ($S1$) of type *contains*. Subsequently, $V1$ and $A2$ are substituted by a place $S2$ of class *equals*. Then, $S1$, $S2$, and $A3$ form a place $S3$ of *meets* type. Finally, $S3$, $I1$, and $T1$ are represented by a place $S4$ of type *equals* [2].

A document can be retrieved by the combination of the following four imprecise matching conditions; (i) temporal condition which specifies the temporal relations that exist among some component media objects during a certain period time of the document presentation; (ii) spatial condition which describes how the spatial layout looks like during the same period of time; (iii) logical-structure condition which delineate the possible logic-relationship of the component media during the same time interval; and (iv) content condition which describes the desired contents of some of the components.

Queries for Playout Control

This type of queries control the presentations of mono media or a document. It offers playout (may be partial interval) at forward/reverse direction with various speed allowable. User's

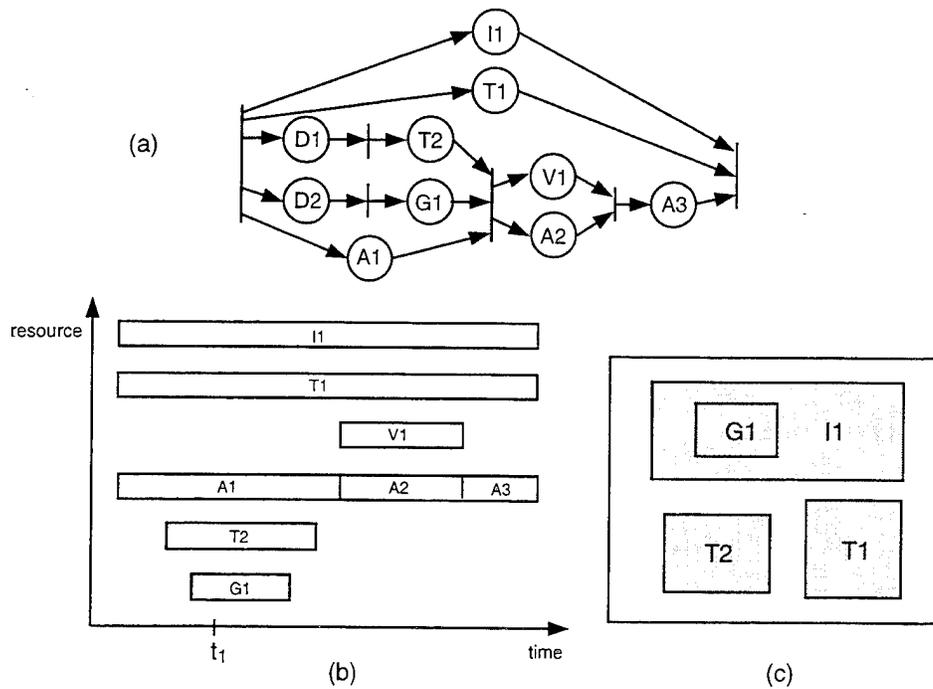


Figure 6: Example OCPN, timeline, and spatial layout at t_1

interactions are also taken into account.

References

- [1] M. Iino, Y. F. Day, A. Ghafoor, "An Object-Oriented Model for Spatio-Temporal Synchronization of Multimedia Information," *Proc. of IEEE ICMCS' 94, International Conference on Multimedia Computing and Systems*, pp. 110-119.
- [2] T. D. C. Little and A. Ghafoor, "Interval-Based Conceptual Models for Time-dependent Multimedia Data," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 5, No. 4, August 1993, pp. 551-563.
- [3] Y. F. Day, S. D. Dagtas, M. Iino, A. Khokhar, and A. Ghafoor, "Object-Oriented Conceptual Modeling of Video Data," *Proc. of IEEE International Conference on Data Engineering '95, Taipei, Taiwan*, pp. 401-408.

- [4] Y. F. Day, S. D. Dagtas, M. Iino, A. Khokhar, and A. Ghafoor, "Spatio-Temporal Modeling of Video Data for On-line Object-Oriented Query Processing," *IEEE ICMCS' 95 Proceeding*, pp. 98-105.
- [5] Y. F. Day and A. Ghafoor, "An Object-Oriented Query Language for Multimedia Database Systems," *Technical Report TR-ECE 95-16*, Purdue University, June 1995.

Agent-Oriented Knowledge-Based Distributed Economy: Abstract

Piotr J. Gmytrasiewicz

Department of Computer Science and Engineering
University of Texas at Arlington, Arlington, TX 76019-0015
piotr@cse.uta.edu

July 1, 1995

The National Information Infrastructure is characterized by being highly distributed, heterogeneous, dynamic, and comprising of a large number of autonomous nodes. We found it useful to view a distributed information environment, such as NII, as a knowledge-based economy. The economic agents populating the environment can be naturally identified either as the suppliers, being in charge of knowledge and data repositories, as the consumers, seeking the information for their own use, or as a mixed consumer/provider type. Using this view, the agent-oriented information services will be able to take the initiative in bringing new, critical material to the attention of the other agents, knowledge repositories and to human users.

An important characteristic of such autonomous agents is that they should be viewed as profit-driven and selfish optimizers. Thus, they maximize their own payoff, and do not directly consider the welfare of other agents when deciding on action or strategy to adopt.¹ In the distributed information economy the agents' actions of requesting, supplying and displaying information are driven by the expected utility of the information and the way in which it is presented.

The main issue that needs to be addressed, therefore, is the method of evaluating the value of information, and its presentation, from two points of views. First, from the point of view of the consumer of the information: What is the expected value of the information, should this information be requested, and if so, what price is the consumer willing to pay? Second, from the point of view of the supplier of the information, what is the value of

¹The agents' preferences will, of course, reflect the interests of the human users or organizations they represent.

transmitting the information to the consumer, should the request be granted, at what price, and in what form should the information be supplied?

In some contexts, for example in virtual manufacturing, it may be useful to view the agents as being comprised of two related components – the production (or action) part, and the information processing part. The production/action part is the one traditionally associated with any company or a person. The information part is one that is expected to direct the production or action activities, say, based on the current state of information about the market conditions, about the other economic players, and other relevant factors. We expect that the way of managing information, by sharing what should be shared with other economic agents, and requesting the relevant information from the others, will have a profound effect on the agents' ability to ally themselves within virtual enterprises best suited to the demand and economic climate, and, ultimately, on their ability to generate profits.

The profits can be seen as generated either by the production/action component, by the information component, or by any combination of the two. The proportion of this mixture determines whether the information is treated as an aid in production-related decisions, when profits are generated by the production component, or whether information is itself sold or processed for profit.

Value of Information

Our basic premise is that the value of information, as well as the optimality of the way in which it is presented, should be the primary factor determining the way to best manage the information at hand. As we mentioned, two facets of value have to be considered. First is the value of the provided information to its consumer. This is the usual notion of the information value considered in decision theory [3]. It is defined as the difference between the expected utility of consumer's acting with the information and the expected utility of action without it.

The second factor is the value of the information provided to the consumer, but viewed from the point of view of the provider of the information. The above two factors are, in general, not equivalent. In an environment in which the preferences of the agents are aligned these two are likely to be similar. If, however, the preferences are conflicting, the two values are likely to be very different. In the economically realistic cases of partial overlap and partial conflict of interest of the parties in question, the optimal information management has to take into consideration the subtleties of what information could be shared (or sold) due to its value a consumer, what should remain private due to its sensitive character to the provider itself, and what information should be requested from other units that are, say, considered as candidates for partners in a virtual enterprise.

The value of the information from the point of view of its provider can be defined,

again, as the difference between the expected utility, *from the provider's point of view*, of consumer's acting with the information, and the expected utility, for the provider, of consumer's action without it. This definition has been used by us in our work on rational communicative behavior [2]. Our method of computing the above value, based on the Recursive Modeling Method [1], gives the agents the ability to model the knowledge states of other agents to predict their behavior. By modeling the ways in which a communicated information changes the other agents' state of knowledge, their resulting expected behavior can be predicted. The desirability of the new behavior, compared to the one without the information, thus yields the value of providing the information.

According to our approach, therefore, the information-related decisions always take into consideration the expected use and decisions the information will facilitate. This lets us assign value to communicative actions, and treats the information as knowledge, as opposed to mere data.

Presentation Issues

Just as the content of a message to be transmitted is subject to optimization based on its value, the form in which it is presented is an optimization decision as well. The fact that an object can be verbally named, as its location is pointed to at the same time, as in "Watch out for the bandit" spoken message accompanied by a light in the head-up display of a pilot's 4 o'clock, optimizes the expected impact of the message, measured as the ability of the consumer to act on it. Our current work attempts to use the modeling methods we previously developed to optimize the allocation of the presentation media to an information content, given the particular decision-making situation the consumer of the information is in.

References

- [1] Piotr J. Gmytrasiewicz, Edmund H. Durfee, and David K. Wehe. A decision-theoretic approach to coordinating multiagent interactions. In *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence*, pages 62–68, August 1991.
- [2] Piotr J. Gmytrasiewicz, Edmund H. Durfee, and David K. Wehe. The utility of communication in coordinating intelligent agents. In *Proceedings of the National Conference on Artificial Intelligence*, pages 166–172, July 1991.
- [3] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufman, 1988.