Perspectives of Multimedia Systems

Reports from the 1994 Dagstuhl Multimedia Seminar

Edited by
Ralf Guido Herrtwich
James D. Foley
José L. Encarnação

Georgia Institute of Technology
GVU Center & College of Computing

Preface

In July 1994, leading international multimedia researchers met in the International Computer Science Research Center at Dagstuhl Castle to discuss the fundamentals and perspectives of their field. The purpose of the seminar was twofold: to arrive at a common understanding of basic technologies of the field as they have evolved over the last decade and to decide on the most important issues for multimedia research in the years to come.

This report provides a summary of the presentations and discussions at Dagstuhl. Enclosed are also the position papers submitted by the workshop participants. It covers a broad range of topics: multimedia encoding methods, operating system support, network and communication technology, storage and databases, mailing, conferencing, and human-computer interfaces. The seminar devoted one session to each of these topics. A so-called white paper presentation introduced the state of the art in each area and provided the basis for a round of discussions that were initiated by position statements from selected speakers. At the end of each session, a research agenda was compiled to collect questions that the seminar participants believed to be of particular importance to the advancement of the field.

At the end of the seminar, a spontaneous poll identified three items as the most pressing issues of multimedia research:

- How to adapt multimedia applications dynamically and continuously to their environment to make them deliver the best possible service under any given set of conditions?
- How to derive and utilize content information within multimedia streams so that query operations can access not only textual indices, but the multimedia information directly?
- How to define scalable mechanisms that can cope with the large volume of multimedia traffic in environments with large numbers of users, all with heterogeneous requirements and capabilities?

But many more research questions have been raised during the seminar and are compiled in the session reports.

It is a pleasant duty to thank all people who have been involved in making this Dagstuhl seminar a success. Thanks go to all seminar participants for lively and at times heated discussions as well as to the session reporters who have captured these discussions for this report. We are particularly grateful to Doris Meschzan who assisted us with the seminar arrangements in every possible way from the first invitations to the final report publication.

Ralf Guido Herrtwich
James D. Foley
José L. Encarnação

Table of Contents

Part I: Workshop Reports

Part II: Position Papers

Part I:
Workshop Reports

2 Media Encoding and Compression

Rüdiger Strack, Fraunhofer-Gesellschaft, Germany

2.1 Survey (White Paper Presentation by Ralf Steinmetz)

The demand for the handling of visual and audio information is increasing at a rapid pace in diverse application fields. Efficient representation of the information is required within all areas both for storage and transmission. Various compression techniques have already been established in order to reduce the amount of data necessary to represent the information. The techniques are in part competitive and in part complementary. While some are already used in today's products, others are still undergoing developments.

The requirements on compression techniques posed by various application areas are manifold. The most demanding can be characterized by the terms low delay, high quality, intrinsic scalability, low complexity, and efficient implementation. Drawing a distinction between conversational (dialogue) and retrieval mode (services) the requirements concerning compression techniques can be described briefly as follows: Both modes require the independence of frame size and video frame rate as well as the synchronization of audio, video, and other media. In addition, dialogue mode requires compression and decompression in real–time and an end–to–end delay less than 150 ms. Fast forward and backward data retrieval as well as random access within 500 ms are required in retrieval mode.

Coding as a field can be subdivided into channel coding and source coding as two subdisciplines. Channel coding focuses on the adaptation of compression schemes to the communication channel. To achieve various QoS (e.g. improvement of error handling) channel coding may introduce redundancy. Source coding can be either lossless or lossy. It is called "entropy coding" if it is lossless and tries to produce a bitrate that is close to the entropy (i.e. minimizes average codeword length). Examples are run–length coding, Huffman coding, and arithmetic coding. Examples for lossy compression techniques are prediction based coding (e.g. Differential Pulse Code Modulation (DPCM)), coding by transformation (e.g. Fast Fourier Transformation (FFT), Discrete Cosine Transformation (DCT)), layered coding (e.g. bit position, subsampling, sub–band coding), and vector quantization. Hybrid coding is defined as the combination of different coding techniques.

The most relevant compression techniques which are in use today combine different coding techniques. Thus, they can be classified as hybrid coding techniques. Taking a closer look at the ISO/IEC and ITU–T standards JPEG (still image), MPEG (video and audio) and H.261 (video) as well as the proprietary standard DVI (still image, audio, video) the following observations can be made: JPEG must be considered as the future standard for coding of still images, due to its variety of alternative modes with high flexibility. Software as well as hardware realizations for the JPEG baseline mode are widely available. H.261, an established standard by the telecommunication world, was dedicated for ISDN usage (p=1,..,32). It addresses conversational services (video telephoning and conferencing) supporting very restricted resolution modes (Common Interchange Format (CIF), Quarter CIF (QCIF)). MPEG is the most promising standard for future compressed digital video and audio. MPEG–1 was optimized for multimedia applications that are based on the retrieval mode. It defines both video and audio compressed data streams offering data rates up to 1.5 Mbit/s. The quality of MPEG–1 video (1.2 Mbit/s) can be compared to VHS–video. MPEG–2 will allow for TV and HDTV quality at the expense of higher data rates (2–100 Mbit/s). MPEG–4 will provide for very high compression ratio encoding of video and associated audio (less than 128 Kbit/s). This may be used for mobile communication. Hereby model–based coding may play a crucial role. DVI, as an proprietary standard, defines still image, audio and video compression. For still images a configuration of JPEG is provided. For video encoding both a symmetric and an asymmetric mode are supported. The latter provides video quality comparable to MPEG–1. Today, many available DVI–implementations suffer from a (de)compression delay above 150 ms.

The technical quality as well as the market availability determine the techniques that will be used in future multimedia systems and services. A cooperation and convergence of the different techniques can be expected. This may include the usage of fractal and model–based coding techniques.

The discussion of the white paper can be summarized as below:

- Other standards: The ISO/IEC standard JBIG should be mentioned in regard to lossless compression for still images.
- Influence of packet loss: Packet loss depends on the error characteristics of the underlying channel. Although different groups are currently working on metrics addressing the resulting image and audio quality there is no acceptable metric available. ATM cell loss rate was mentioned as one example for a metric.
- MPEG–1 data rate: The channel has a constant data rate while there is no constant rate from the logical point of view (bits to compress different Groups of Pictures (GOPs) differ). However, there is no enforcement to produce a constant channel rate unless specific hardware requirements hold. A constant channel rate

can be achieved by using buffering mechanisms. The buffer can be moved across the network.

- Degradation: Concerning degradation the two aspects distortion and network degradation have to be distinguished.
- Audio compression: 1.4 Mbit/s are used to store audio on CD. Thus, to store audio and video on CD compression techniques have to be applied to both representation media. According to tests made, MPEG–1 audio compression with 384 kbit/s (factor 4) achieves acceptable i.e. transparent quality.

2.2 Position Statements

2.2.1 Bernd Girod

In order to cope with the various constraints in regard to access rates, network bandwidth, and storage capabilities video as an integral part of multimedia systems has to be compressed substantially. However, highly efficient compression as a necessary prerequisite for the storage and transmission of video conflicts with several other requirements including scalability, support for interactive video, and editing capabilities. The term “scalability” encompasses three issues:

- Image size scalability: The spatial resolution of the video frames should be flexible according to the specific quality required by the user/application.
- Partial decodability of a compressed data stream: The receiver should be able to decode and display an image from partial information. Hereby, image quality should degrade gracefully. This issue is especially addressed by the digital broadcasting area. Within multimedia systems it is e.g. of specific relevance if data (audio, images, video) is transmitted over networks without guaranteed Quality of Service (QoS).
- Computation–limited coding/decoding: The computational bandwidth that affects coding and decoding should be scalable in such a way that the same compressed data stream could be coded and decoded with processors of different power, e.g. by de–/increasing image quality.

Evaluating current video compression standards concerning scalability the following observations can be made: H.261 offers no scalability. MPEG–1 offers a kind of temporal scalability in such a way that bidirectional prediction pictures (B–frames) can be left out without hurting Intrapictures (I-frames) or Predicted pictures (P–frames). However, no mechanisms for spatial scalability are provided. MPEG–2 is not scalable by nature. However, compression schemes can be built with the MPEG–2 toolbox that address scalability, both spatial and temporal.

Concerning the support for interactive video, VCR features (e.g. shuttle services) and random access should be supported. Furthermore, a short decoding delay should be provided. However, there is a trade-off between the bit rate and the decoding delay. Low decoding rate and short decoding delay can not be achieved at the same time. Thus, we can for example not expect that MPEG–4 will have a short decoding delay.

Concerning video editing interframe and intraframe coding have to be distinguished. While interframe coding requires the decoding and encoding for each editing process, intraframe coding (e.g. M–JPEG) is preferable for editing. The latter provides editing on the data stream level.

Although many compression requirements have been already addressed at least partially, the big challenge remains and can be still characterized by the terms highly efficient compression, scalability, support for interactive video, and editing capabilities.

In the context of editing it was stated that it might not be adequate trying to push compression issues into research areas that are not directly addressed by the compression research community.

During the discussion, the question occurred what is actually meant by using the term "editing"? Editing in the above sense addresses video post–production. Different aspects concerning editing were discussed. It was stated that for applications in general it might be appropriate to abstract from the concrete external data representation. In this context, transparent disk compression was mentioned. While the application operates on uncompressed data it is stored transparently in a compressed form. Although the example was based on lossless compression, this may be also feasible for lossy compression insisting that only the first pass is lossy.

The issue of object recognition was discussed. Although this issue is mainly addressed by the image processing and computer vision community, it has to be dealt with for the establishment of model–based schemes.

2.2.2 Larry Rowe

Near term trends in the area of video encoding and compression can be expected as follows:

- MPEG–1 will be used on every desktop: Both chips and boards will be available at very low costs. Thus, MPEG–1 will be available on any platform (PC, workstation, etc.). The demand for playback application will increase.
- M–JPEG will be used for editing systems: Both chips and boards will be available at very low costs and there will be a large installed base of M–JPEG applications. The problem that M–JPEG is currently not compatible with MPEG I–frames (e.g. Huffman tables differ) may be solved by the development of a JPEG–2 compression technique.
- H.261 will be used for conferencing: However, the question occurs how long it will be still used due to the availability of MPEG–1 chips and boards at low costs?

Looking more precisely on research issues for MPEG–1 the following issues can be identified:

- To support video editing existing software and hardware (chip specific) interfaces should be improved. Moreover, frequency domain operations should be further developed to increase editing performance. Nevertheless, the editing software itself should abstract from the respective internal data representation.
- To conceal errors source/channel coding should be improved.
- High–quality encoders today are very expensive. Therefore, low–cost encoders should be developed supporting simplified functionality and quality.
- Perceptual coding models should be established to improve the compression ratio. For example, many compressed information could be thrown away changing from light to dark scenes.
- The future architecture of codecs should be aligned with the resulting benefits comparing customized–chips vs. general purpose processors as well as multiple CPUs vs. special co–processors on chip.

Beyond MPEG the following requirements and expectations can be identified:

- Within the area of mobile computing there is a demand for a different kind of compression. First, you need low power algorithms. Thus, decompression at the receiver should be performed with minimum computing requirements (e.g. vector quantization). Second, the reliability of the transmission depends on the power used to transmit the data. To optimize the reliability, the signal may be split into a high priority channel (sent with high power) and a low priority channel (sent with low power) and merged together at the receiver.
- The usage of source/channel coding for mobile computing requires scalability and priorization support.
- Fractals/wavelets will not gain major market share. Although e.g. the usage of wavelets may have benefits compared to the DCT, these are not significant to be able to compete with the existing DCT–based market.
- MPEG–4 at very low bit rate may be useful — in a modified mode — for mobile computing.
- Other (de facto) standards will vanish, e.g. CellB, QuickTime, etc.

The future usage of H.261 was discussed. Currently, ISDN is the only wide area network that is broadly available and suitable to transmit video. Thus, H.261 may be still needed. However, within corporate networks the usage of M–JPEG already works reasonable well (e.g. supporting digital video with QCIF/CIF resolution). Besides, chips and boards supporting JPEG are cheaper.

The future usage of de–facto standards was discussed. Existing international standards work well for full color images. However, they do not address 8–bit look-up table (LUT) images. Thus, some de–facto standards still may be used.

2.2.3 Wolfgang Effelsberg

The standardized video compression techniques that are in use today (e.g. H.261, MPEG, M–JPEG) require much computation. They were developed under the assumption that specific hardware for video (de)compression is available. For the time being a number of problems derive from the usage of (de)compression hardware, e.g. the dependency of hardware compression boards on bus and graphics display, the missing flexibility to perform operations within the (de)compression process, etc. Evaluating compression ratio versus universal availability and flexibility the latter will become more important. Therefore, as a general guideline video (de)compression should be performed in software. Software (de)compression — as an integral part of future window systems — will enable users to participate in the multimedia world, independently of their platforms.

While compression techniques that are based on the DCT are well understood and are already used in various application areas, new and very promising techniques are still under investigation:

- Fractal compression techniques for images and video
- Usage of wavelet transformations instead of the popular cosine transformation
- Usage of non–linear characteristics of the human eye for more efficient video/ image compression as used for audio–compression that is based on a mathematical model of the cochlea.

Further research has to be performed within these areas. Moreover, as long as 8–bit graphics adapters still dominate the market dithering aspects should be considered for the establishment of new compression algorithms. The same holds for supporting the integration of algorithms and tools (e.g. cut and edge detection) in the middle of the decoding process.

Also, lossless compression should be addressed more explicitly since many application areas like remote sensing, medicine, etc. prefer or need to deal with the data as originally acquired.

A discussion concerning the realization of compression techniques in software and hardware took place. It was stated that the application should not have to care about the realization. A software solution might be adequate to integrate (de)compression into MM–extensions of the operating system. However, such an integration in general may not be feasible due to the large number of different (de–facto) standards available.

The distribution of compressed data streams was discussed in the light of compression "units" and efficiency. Different modes of distribution in regard to MPEG–1 were distinguished. DCT–blocks, slides (some number of macroblocks across some images), frames, and groups of frames may be distributed. Further research has to be performed in this area.

The question of performing video compression on a general–purpose CPU was addressed. In this context, the problems in regard to time slicing and the establishment of a “compression description language” were shortly discussed. The latter focuses on an open language that is not based on a specific compression technique addressing elements common to all data streams. Such a language may be for example used to store compressed data in a technique independent manner while supporting requests for specific data representations (e.g. MPEG, H.261, M–JPEG, etc.).

2.3 Research Items

Although a well–defined set of techniques for (de)compression has been developed already the challenge within the area of media encoding and compression remains. The questions and items for future research in this area cover various aspects regarding the flexibility, performance and usability of compression techniques as summarized below:

- Compatibility and convergence of compression techniques: For the establishment of future compression standards, compatibility to existing and forthcoming standards should be considered to assure efficient handling of compressed data independently on the application area. For example the problem that M–JPEG is currently not compatible with MPEG I–frames could be solved by the development of future compression techniques for static images.
- Support of video editing: How can video editing be supported more efficiently?
- Interfaces to compression hardware (i.e. compression chips) and software: Well–defined interfaces have to established.
- Frequency domain operations should be further developed to increase editing performance.
- Codecs and presentation: Should future codecs be distributed together with the display/presentation tool? Dithering aspects should be considered for the establishment of future compression techniques as long as 8–bit graphics adapters still dominate the market.
- Potentially high compression coding techniques (e.g. fractal compression, model–based coding, etc.) should be examined and their applicability for particular types of application analyzed.
- Further lossless compression techniques should be established that are applicable for different application areas (e.g. remote sensing, medicine, etc.).
- Fractal compression: The usage of fractal compression techniques for still images and video should be further investigated.
- Wavelet transformations: What are the benefits of the wavelet transformation compared to the cosine transformation and how can it be used for more efficient video compression?
- Perceptual coding models should be established to improve the compression ratio.

- Investigation on the non–linear characteristics of the human eye: While audio compression makes usage of a mathematical model of the cochlea based on the non–linear characteristics of the human ear, similar characteristics of the human eye so far are not very well understood.
- Development of low–cost encoders supporting simplified functionality and quality.
- The future architecture of codecs should be aligned with the resulting benefits comparing customized–chips vs. general purpose processors as well as multiple CPUs vs. special co–processors on chip.
- Distribution of compressed data streams: How can compressed data streams be distributed efficiently to different processors? What are the appropriate “units” to be distributed?
- Establishment of a “compression description language”: An open and extensible language should be developed that addresses elements common to all data streams. Such a language may be for example used to store compressed data in a compression technique independent manner while supporting requests for specific data representations.
- Scalability and priorization support: Scalability and priorization support are necessary prerequisites for mobile computing.
- Support of image size scalability: The spatial resolution of the video frames should be flexible according to the specific quality required by the user/application.
- Support of the partial decodability of a compressed data stream: The receiver should be able to decode and display an image from partial information. Hereby, image quality should degrade gracefully.
- Computation–limited coding/decoding: The computational bandwidth that affects coding and decoding should be scalable in such a way that the same compressed data stream could be coded and decoded with processors of different power, e.g. by de–/increasing image quality.
- How should the different coding elements be packetized into network packets?
- Influence of packet loss: Packet loss depends on the error characteristics of the underlying channel. To conceal errors source/channel coding should be improved. Also metrics should be developed. Currently there is no acceptable metric available addressing the resulting image and audio quality.

3 Abstractions for Multimedia Computing

Martin Frühauf, Computer Graphics Center (ZGDV) Germany
Maria Perez-Luque, Boston University, USA

3.1 Survey (White Paper Presentation by Thomas D. Little)

Abstraction is used in many aspects of computing. It is particularly appropriate for multimedia computer systems due to the number of system components that must be integrated to yield working applications. Various perspectives that lead to abstraction can be found: the user, the application developer, the system integrator, and the system component developer.

Many abstractions have been introduced in all areas of multimedia computing. There are abstractions for individual data, multimedia integration, media manipulation, operating system support, communications, databases, distributed system architectures, etc. These abstractions have the following advantages:

- Multimedia technology is diverse. Abstraction is a useful technique that can help a user, a programmer, or a component designer deal with this system complexity.
- Abstraction is very useful to isolate technical problems, e.g., data compression techniques from the applications programmer.
- Once the components/services are partitioned and interfaces are defined, it is possible to compare and evaluate the components/services and to reuse the developed technology.
- The separation of object definition from rendering defines a framework for the measurement and control of quality of service.

Nevertheless, there are still a lot of problems to solve. Some of them are related to the already proposed abstractions. But the most difficult part is to study the connection among all the already defined abstractions. Abstraction does not necessarily mean compatibility. A lot of work has to be done to achieve a good definition of individual abstractions and their integration in the development of systems.

The points of the discussion of the white paper on abstractions for multimedia computing can be summarized as follows:

- Compatibility of abstractions. A lot of abstractions have been defined, and a lot more will be. Most of them are incompatible. Do we want to generalize abstractions?
- Utility of abstractions. We have to find the real utility of abstractions. Abstractions are not the goal but the means. We need to see how abstractions can help solve difficult problems.
- Definition and creation of abstractions. The process of definition of abstractions was also discussed. Abstraction is an ongoing process. Consider the evolution of programming languages, in which the level of abstraction has been increased over the last decades from machine code to assembler code, functional programming languages, and the object-oriented programming paradigm.

There is still no agreement about the right process of defining abstractions: create the system and find the abstractions, or define the abstractions and then create the system.

- Abstractions of the future. There was concern about the abstractions that everybody will use in the future, in comparison with the abstraction that are currently used e.g., stream abstraction. Among others, the following were suggested:
- Quality of Service (QoS). Why are QoS so difficult to provide? Is it a matter of finding the right abstractions for multimedia computing systems? What is the relation to open distributed processing?
- Generalized information interfaces. Is an “information interface” an abstraction that can be used for the design of human-computer interfaces during the development of new multimedia information systems?

3.2 Position Statements

3.2.1 P. Venkat Rangan

Venkat Rangan presented three kinds of abstractions, which are at different levels, used for the multimedia server developed at the University of California at San Diego:

- Service Abstractions (service level):
  e.g., cable TV, video-on-demand, virtual VCR, personalized video channel
- Semantic abstractions (system level):
  + content: e.g., shot, scene,
  + accessing video and content-based retrieval of multimedia information
- Syntactic abstractions (system level):
- media units: e.g., video frames, audio samples, in general: streams

Each individual media has these three levels of abstraction.

It is difficult to go from one level of abstraction to the other, specially, how to pass from the syntactic level (e.g. storage of media) to the semantic level (e.g. content-based retrieval).

Abstractions on the service level:

Abstractions on the service level must reflect the way users interact with the system and the system services. There are a lot of abstractions for structured interaction. But, it is very difficult to identify abstractions for unstructured interactions; e.g., what are paradigms for content-based retrieval of multimedia.

Abstractions on the system level:

What are the right semantic abstractions? Possibly an information abstraction: Building abstractions of the material used in a multimedia information system; but, building the right abstraction of an information depends heavily on the area of application (e.g. entertainment vs. education).

Identifying the right abstractions on the semantic level for the storage and the retrieval will be very helpful for the design and the development of content-based retrieval of video.

In the beginning the discussion focused on the proposed abstractions on the service level for a multimedia server. A question asked was whether it is the right approach to develop a digital VCR for the delivery of video-on-demand

Nowadays, customers are used to handle the user interface of a VCR. Because little knowledge and experiences in the area of content-based retrieval of video data is available, it is hard to identify today the right abstraction for that purpose. When we know something about video content we will be able to improve the system and to define an abstraction for content-based retrieval.

But, one of the open problems is if developing systems that support content-based retrieval of video data is the right approach to add value to video-on-demand services. Scenario: Usually a person sits and watches TV. What is happening today is switching channels too often using the remote control if the material is not interesting enough. People do not want to push buttons and search what is available on interactive TV. It is just: If I like it, I want to watch it.

So, what is interactivity for while watching TV? Providing an abstract of the material that is available for a quick overview may be helpful for the user to choose the right channel. Thus, techniques for compression in time of the content of the material are a key technology, i.e. extracting the essence of a movie or a document.

Providing an abstract for each movie or other video increases the difficulty and complexity of creating multimedia Information. It is much better, but it takes a lot of effort in authoring. So, what about the costs of content acquisition to add some value to video-on-demand?

Then, the focus of the discussion changed to the abstractions on the system level (semantics and syntax).

It was discussed whether there are common syntactic level abstractions that are universally used. Are streams such an abstraction?

An example for syntactic level abstractions could be the structure of a document in a storage system. Is it worthwhile to separate between time and other types of layout information in the definition of the structure of a multimedia document? This raised the question what abstractions for layout of information presentation really are? Possibly, just a position, a point or an entity in a space of n-dimensions.

The discussion on the layout of information for presentation raised the question on the role of metaphors in multimedia systems. Intuitive metaphors are powerful abstractions. They are a useful service for the user of a system to help to understand the purpose and the behavior of the system. But, providing good metaphors is really a design problem, it is not a scientific problem. More or less the user perspective of a system is represented in the metaphor. Intuitive or real-world metaphors are the best abstractions from the user's point of view. A file system metaphor, on the other hand, is an abstraction suitable for the system developer.

But, first of all it has to be defined what a metaphor and what an abstraction really are. What is the difference between abstractions, paradigms and metaphors?

3.2.2 Sape Mullender

Sape Mullender's view on a true multimedia system is that true multimedia must include the capability for programmers and users to process the information encapsulated in all the different media. The structure of a multimedia system should be such that, once we know algorithms for processing information contained in video streams, we can sit down and write programs. The support of audio and video requires systems to have a strong notion of timing - of performing not only the correct actions in the correct order, but also at the correct time.

To achieve this and to realize such a true multimedia system, a huge number of technical problems has to be solved, first, which have nothing to do with abstractions. Abstractions are good for helping the understanding of the systems, but they are not the goal. The goal is to build the system. The experience derived from building the system is much more helpful for understanding the system than abstractions of a system which has not been built.

As the main part of the presentation focused on real system design and not on abstractions, also the discussion turned to focus on the project presented, specially on the ATM architecture.

3.2.3 Doug Shepherd

Doug Shepherd's initial comment was that the worst (but most interesting) things in computer science include solving hard problems and complexity. The goal of system designers is to simplify problems. Abstractions simplify problems. A complex problem is divided into a sum of simple problems that can be solved.

Abstractions should come up from experimental implementations. His experience is that a lot of problems appear when the other approach is taken, i.e.when one moves from abstractions to implementations.

Multimedia system support and abstractions currently used or needed include:

- Streams.
- Traditional kernel interface for read and write; better for multimedia would be a passive role of the application and letting the system itself take responsibility for initiating events.
- RPCs (Remote Procedure Calls) are not a good abstraction for multimedia, an abstraction to replace RPC is necessary.
- Filters; better named “adaptors” because “filter” suggests information reduction.
- File servers are an example for an abstraction that reduces the complexity of the device itself.
- QoS (Quality of Services); better requirements on the end system because QoS are not really defined yet; QoS parameters of relevance are generally agreed to be bandwidth, latency and error rates but not jitter.
- Metadata description languages (very important, e.g. for content-based retrieval)

At the beginning the discussion focused on the type of information to store as metadata information. The location of information, the prices for retrieval as well as the quality of services available have been suggested to be stored as metadata. Metadata information in general should contain information on how the data itself has to and can be handled. This discussion led to the fundamental question of the difference between a metadata description language and the structure of multimedia information.

Next, the discussion focused on quality of services (QoS). QoS is beyond transmitting video from A to B. In addition QoS is communication and interaction between users, QoS is synchronization, QoS has to be defined for all kind of media.

As far as abstractions are concerned it has been claimed that abstracting from QoS is the wrong approach. QoS is a problem with two dimensions, i.e. QoS depends on the stuff (the type of media) to be handled and the class of application of a multimedia system (e.g., computer supported cooperative work (CSCW), video-on-demand, information kiosk).

3.3 Research Items

The questions and items for future research reflect many of the open and unsolved problems of multimedia computing and system development for multimedia computing. Abstractions are meant to help to solve these open problems. Questions on abstractions from the following categories were identified:

- Distributed systems, i.e. peer-to-peer connections and client-server architectures
- Multimedia databases
- Interoperability and media exchange
- Quality of Services
- Presentation and synchronization

As far as abstractions for distributed multimedia systems are concerned, the following questions need to be considered:

- How can the network architecture (peer-to-peer vs. client-server) be abstracted for the application programming interface (API)? How can the system, not the application, adapt to the addition and deletion of participants (i.e., reconfigure to a server-based solution from a peer-to-peer configuration)?
- What fundamental services are necessary to support various classes of distributed applications? E.g., support for separate multicast and unicast, shared resources (file locking, screen locking, shared pointer), token passing (shared pointers), floor control, activity logging, and topic indexing.
- What abstractions should be chosen to reduce the maintenance cost for a distributed system?

- What are appropriate domain information models for supporting various application domains? What are their canonical forms, if any? How can they be designed to permit database interoperability for DBMS operations such as searching?
- What is the abstraction vs. performance penalty in object-oriented multimedia database systems? What is the interoperability penalty? (E.g., window systems, operating systems, database systems.)

- Can we converge on a standard set of data formats that support scalable multimedia services, media conversion, hypermedia, etc., over a wide range of platforms?
- How should the media manipulation be defined? What models or languages should be used?
- How can compatibility be achieved for object-oriented frameworks developed from different abstract models?

- How is QoS characterized and modelled?
- How does the programmer specify a range of QoS?
- How is an object's method performed to achieve a specified QoS?
- What are the relationships among operating systems, communications, and databases with respect to QoS? What type of abstractions can we use to describe that?

- What abstractions on the various media can be used to support fast browsing of very large information spaces?
- What is the appropriate level of abstraction for the specification of presentation timing requirements? What is the elegance vs. efficiency trade-off?

Most of the people in the audience seemed to be more interested in solving particular problems of development and implementation of multimedia computing than in building abstractions for multimedia computing systems. The “bottom-up approach”, i.e. solving a number of specific problems to build a system, was favored above the “top-down approach”, i.e. building an abstraction for the entire system as well as for subsystems or system elements followed by the development and implementation of the components. But, there are several questions that appear more or less interesting:

- How to define abstractions for QoS at all systems levels?
- Definition of generalized information interfaces.
- Compatibility among abstractions:
- There are abstractions for the same system level (or data). How to connect abstractions of the different part of the systems?

From the application and users point of view seem to be two concerns:

- Finding good “real-world” metaphors for different applications of multimedia systems, and
- application abstractions, or general services abstractions to construct different applications.

However, it could be observed that there is a lack of clear definitions for a number of terms used in multimedia computing, e.g.:

- What is QoS?
- What is content-based retrieval?
- What is metadata?
- What is a metaphor?
- What is the difference between abstractions, paradigms and metaphors?

Abstractions are definitely helpful to formulate the missing definitions.

3 Multimedia Storage and Databases

Gerold Blakowski, IBM, Germany
Srihari Sampathkumar, University of California at San Diego, USA

3.1 Survey (White Paper Presentation by Desai Narasimhalu)

Multimedia technology is a seamless integration of monomedia technologies such as audio, video and text and provides interactivity as well. A multimedia database management system (MMDBMS) is a tool for efficient organization, storage and retrieval of multimedia objects (MOBs).

There are three fundamental differences between MMDBMS and traditional database management systems (DBMSs). The first factor is that multimedia data is audio-visual in nature and accurate representation and querying of audio-visual data is still a challenge. Secondly, traditional DBMSs store and maintain attribute based data. Content extraction and representation from MOBs is still a challenge. Lastly, transcoding, which is representation of the same information in different forms is a new factor and the MMDBMS has to support multiple representations of the same data.

The above factors pose a significant challenge and were discussed in the session on Multimedia Storage and Databases that is summarized in this section.

Database Data Models

Traditional models of databases have been restricted to hierarchical, network, relational and inverted file. The more recent models have been entity-relationship, object-oriented (OO), temporal and spatial. The more common model is the OO model.

The reason for lack of standards in the OO model is partially due to the existence of two different approaches. The first approach is to extend other data models or define a new data model. For example, in Postgres, the relational data model was extended to integrate the OO paradigm in it. The second approach is to use OO languages such as C++ as the basis for defining a data model like the Versant OO DBMS.

The database creation process in general follows the steps of preprocessing, segmentation, classification/clustering, indexing and storage. In traditional databases, all but the indexing and storage steps are handled manually.

Models for a MMDBMS

The building blocks of a MMDBMS are based on models for data/object, transaction, query, storage, and interface.

The multimedia object data model is comprised of three layers, one for defining data types (compound video, image, text, etc.), the middle for defining object types (logical, compound, fuzzy, etc.) and the top layer for defining relationships between objects (spatial, temporal, inverse, etc.).

The transaction model has a manager at each layer, with the first layer for capturing concurrency control schemes, the second for representing locking mechanisms, the third for handling alternative approaches to updates, the fourth to maintain version control mechanisms and the last layer for handling integrity enforcement.

The query model has three layers with the innermost layer for representing a portfolio of the query engines, the middle layer for representing the subqueries from a compound query and the query evaluation plan, and the outermost layer for representing compound queries and their results.

The storage model has four layers, namely, the information access that interfaces to the directory, the indexing mechanisms, the buffer management techniques and the I/O models available to the MMDBMS.

The multimodal interface model has three layers: a collection of query interfaces for each mode (text, audio, video) supported in the MMDBMS, a query refinement layer and a query integrator layer in which the query integrator takes as input the queries from the different interfaces and packs them as single query.

MMDBMS Architectural Issues

MMDBMS architecture will consist of five main managers namely, the interface manager, object manager, query manager, transaction manager and storage manager which in turn are supplemented by a thesaurus manager, a context manager, a configuration manager, a data migration manager, integrity rule base and a directory manager. Each of the main managers is built to preserve the integrity of the corresponding models described in the previous section. Each main manager will have a cooperating autonomous intelligent agent which handles most of the communications at the peer and the family level.

Efforts in MMDBMS - Past and Present

Efforts in the past have been limited to new kinds of data, rule processing, data model, tertiary storage, long duration transactions, version and configuration management, semantic inconsistency, site scale-up, image understanding, recognition and interpretation and image database management issues. More recent efforts have focused on memory management, feature-based indexing, query processing, interfaces and applications. Content-based retrieval and video on demand have been the two MMDBMS related hottest research topics in the last two years.

The most comprehensive multiple query and retrieval system hitherto reported runs on an engine called MUDE (MUltimedia Database Engine) developed at the National University of Singapore that is enabled to handle composite queries involving qualifiers or constraints that are in image, fuzzy, free text and standard attribute values allowing browsing of multimedia objects using iconic indexing.

Closing Remarks

The following issues were emphasized and discussed in the white paper presentation: Configurability, reusability and extensibility were described as the three main issues in MMDBMS design. It was discussed that a MMDBMS interface is not just a user interface. It is a consequence of a layered MMDBMS architecture. The real-time aspects in a MMDBMS are not necessarily implied from the real-time nature of continuous media. They are mainly implied by the task of the application that uses the MMDBMS. Another statement was that the main problem in databases access lies in indexing rather than retrieval.

MMDBMS is a confluence of a number of technologies such as information storage and retrieval, cognitive science, neural nets, expert systems, data mining and fuzzy set theory. Applications of MMDBMS include video on demand, mechanical and electrical computer-aided design, home shopping and digital libraries. Some of the related technologies which are not considered as part of current DBMS technology are preprocessing of multimedia objects for content extraction, presentation of compound objects, transcoding and annotation and classification based on semantics.

3.2 Position Statements

3.2.1 Klaus Meyer-Wegener

First, MMDBMS is responsible more for storage management and retrieval and not for the user interface. Thus, their API should be the focus and not the interactive interface. Second, the different media cannot be mapped to a single storage concept. Issues such as allocation of secondary storage and buffer management to name a few, are radically different for media such as text and video streams. The crux is that in the future, there will just be a fileserver storage and the database engine will just use it. The database itself is not the storage here. A fileserver need not know whether it is text or video. So why is it not just one concept of storage? The answer to this question is that a DBMS is different from a fileserver. For e.g., WWW is not a MMDBMS because it is just a tree database without any search and multiuser mechanism. In fact, for MMDBMS, a Unix filesystem cannot be used because no mapping to the disk is allowed. Third, ADTs must be defined for each media object so as to guarantee device and format independence. Fourth, the emerging multimedia network information systems (WWW, gopher, XMosaic) are currently incompatible with MMBDMS because they are just fileservers designed for stand-alone interactive use with the handling of the multiple media left to external applications such as ghostview. Such systems cannot be integrated into larger applications.

The question of the lack of standards for MMDBMS is bound to arise. The positive aspect about the lack of standards is that if a standard proves unsuitable, freedom to switch to another or to just wait for the next one to emerge exists. However, the industry is going to reimplement a researcher's idea according to their standards despite the researchers' efforts to implement a certain set of standards.

3.2.2 Arif Ghafoor

The main issues in multimedia databases are: (1) development of models for capturing the media synchronization requirements, (2) development of semantic models for stored multimedia information (3) design of powerful indexing, searching and organization methods for multimedia data, (4) design of efficient multimedia query languages and (5) development of efficient data clustering and storage layout schemes.

A very important concept in multimedia databases is that the indexing and annotation should not only be indicative of the content of the media but also of their context. Hence, the queries may not only require content-based retrieval but also need evaluation of a concept that may involve the temporal dimension.

Many queries will also need searching of data in one stream associated with the data in another stream. For example, a query such as "Get all the video clips where President Kennedy, during one of his cabinet meetings, has made remarks about the fifth amendment" needs searching of audio data associated with the video data of President Kennedy's cabinet meetings. The processing of queries in video databases involves computations such as the symbolic processing for face and object recognition and tracking the motion of objects in video frames.

3.3 Research Items

In the white paper and during presentations and discussions many research areas have been identified for the multimedia storage and database area. In the following the resulting research questions have been classified into the areas of real-time access, multimedia object representation, data modelling, database mechanisms, user access, content-based retrieval and MMDBMS architecture.

How can real-time access to complex and large multimedia objects be achieved?

- Applications such as Video on Demand which are closely related to MMDBMS need additional storage outside the database for program caching. How can real time demands be fulfilled by the MMDBMS without such additional storage?
- To support real-time and synchronized access to complex compound multimedia objects, appropriate clustering and storage layout schemata are needed. What are efficient cluster and layout schemata for single disk and RAID systems as well as for distributed databases?
- Hierarchical storage systems can be developed with hot storage that allows to immediately access data, warm storage that supports access in the millisecond range and cold storage based on advanced tertiary storage devices for access times of minutes. What are the suitable mechanisms for the transparent migration of data through the three types of storages?
- Clients accessing a MMDBMS have to go through a QoS (Quality of Service) negotiation for a global resource allocation. How can the network and operating system allocation schemata be coordinated with the mechanisms such as buffer management, real time access and admission control in a MMDBMS?
- A multimedia database run-time system must take the different performance characteristics of the multimedia objects into account. What are the different characteristics, how can they be described by the method implementor and how can they be used by the run-time system?

How should multimedia objects be represented in a MMDBMS?

- Multimedia objects can consist of either a single medium in which case they become media objects or can have multiple media streams. Both representations have advantages and disadvantages. Which representation should be chosen for a MMDBMS?
- It is necessary that the user can easily retrieve an entire complex compound object as well as its components. How can the compound multimedia objects be represented to support elegant and efficient retrieval?
- If compound objects are stored as Binary Large Objects it is necessary to store information about their structure and components. How and where can this information be stored?
- Multimedia objects can be represented several time in a database due to transcoding demands. What are the mechanisms that allow simple management and access to the multiple representations?

- Databases need to support a fuzzy representation of objects. For example a database consisting of "interesting places" has to allow for a place stored as "interesting" with some non-zero membership value. How does such a fuzzy representation look like?

What are the suitable data models?

- The expressive power of a data model should allow to explicitly identify spatial and temporal multimedia objects. How can a data model support the representation of both temporal and spatial multimedia objects? In addition, how can the data model support the representation of real-time objects?
- The structure of a multimedia object comprises its attributes, content, behavior and function. The behavior is the set of messages it understands, responds to and initiates. The function is the explicit definition of the logical role of the object in the represented world. How can the whole structure be represented in a data model?
- Beside the traditional relationships like "part-of" and "is-a" a new relationship "similar-to" could be introduced in a data model. Also, some multimedia objects may define crisp classifications. How can a data model support the representation of such fuzzy classifications in the classification hierarchy?
- Other, very special relationships are known between the media objects of a multimedia object, e.g. synchronization, hyperlink. Should the application or the MMDBMS handle such relationships?
- The definition of abstract data types for multimedia is difficult. They may include 50-100 operations. What are the appropriate formal specification techniques for multimedia abstract data types?
- Objects may contain logical and physical representations, like the content layout and a structural (section/subsection etc.) layout in a document. How can the notion of mapping between these representations be captured in a data model?
- Multimedia objects have synchronization requirements for their components. Models for the media synchronization must be developed and integrated into the database schema. These models must be transformed into a metaschema to determine the synchronization requirements at retrieval time. What are the suitable models and meta-schemata and how can they be integrated with higher level information abstractions like hypermedia?
- Conceptual models for multimedia data with rich semantic capabilities are needed to be able to provide canonical representations of complex images, scenes, events in terms of objects and their spatio-temporal behavior. What are the suitable conceptual models for multimedia information?
- Meta knowledge about the stored data can be partitioned into application domain and application task knowledge. What is the relevant meta knowledge for a MMDBMS, how can it be used and how is it stored?

How should the database mechanisms be changed?

- Multimedia objects can be very large. What are the appropriate buffer management policies?

- The semantic of updates and transactions on multimedia data is still an unsolved problem. What are the appropriate logging and locking granularities and techniques and how can nested transactions be supported?
- Several types of transaction management like optimistic and pessimistic as well as event driven or data driven transactions may be needed for different applications. What are the appropriate transaction mechanisms and how can a set of selectable transaction mechanisms be provided concurrently?
- Hierarchical indexing in a MMDBMS has to take into account fuzzy classes. How can we enforce indexing on class hierarchies comprising fuzzy classes?
- Content-based indexing may be based on characteristic values of multimedia objects. Structure-based indexing on the logical structure of objects is another possibility for indexing. In addition indexes for the mapping between physical and logical representations have to be provided. How can these indexing mechanisms be used and what are other appropriate indexing mechanisms?

How can the user access the data and how can content-based retrieval be supported?

- Content-based retrieval on multimedia objects demands for new methods. What do audio-visual queries that support such content-based retrieval look like?
- In non-textual queries, there is a need for the support of audio-visual query refinement techniques. They may be based on relevance feedback. What do such query refinement techniques look like?
- Browsing techniques for the access to multimedia databases must be supported. What are the suitable browsing techniques in a MMDBMS?
- Known techniques for content-based search are not compatible with current techniques (e.g. image analysis, speech recognition) or they are not sufficiently powerful, like keyword descriptions. What new structures for description data can be used for a content-based search, what are the comparison operations in the multimedia abstract data types for this and how can they be handled user-understandably?
- Multimedia applications have to interface and output information into other applications such as image processing software. In image processing software, edge detection is extremely important. Hence, compression techniques which sacrifice the sharpness of edges cannot be used. In contrast, applications with human interfaces, precision in the colors of the picture cannot be sacrificed while the sharpness of edges can be sacrificed to some extent. Again, compression techniques more suited for such applications have to be used. How can the automatic compression and coding mechanisms be made independent of variations in the application interfaces?
- If descriptions are used as the base of content-based search, can these descriptions be created automatically during the capture of data or at a post-processing step?
- The access to multimedia data should be supported by advanced user interfaces. Should these user interfaces be part of the MMDBMS or of the application or

should the MMDBMS provide library support for applications to build interfaces?
- Multimedia objects can be regarded under different contexts, depending on the user of the MMDBMS. How can context-sensitivity be supported?

What are the suitable methods for query processing in a MMDBMS?
- Often users can describe the queries in a vague or fuzzy manner. How can query processors handle such ambiguous or fuzzy descriptions?
- The complex nature and audio-visual nature of multimedia objects and the fuzzy description and classification of objects in queries demand a new definition of the join operation and new query evaluation plans and query evaluation processes. Similarity and ranking techniques have to be used. How and with which techniques does a query processor work in a MMDBMS?

What is the appropriate architecture of a MMDBMS?
- An architecture of a MMDBMS may be integrated or federated. What is the suitable architecture?
- A multimedia database can profit from the use of intelligent agents, e.g., for the mapping between representations and selection of suitable classes. How can a MMDBMS be constructed as a group of cooperating autonomous intelligent agents?
- A media object store demands more functionalities than provided by a file system and less than that provided by a MMDBMS. What are the appropriate architectures and functionalities of a media store?
- MMDBMS research comprises of several other disciplines such as information storage and retrieval, image processing, pattern and speech recognition and fuzzy logic. How do we integrate the results of these disciplines in a MMDBMS?

5 Multimedia Networking and Communication

Mahesan Nandikesan, Columbia University, USA

5.1 Survey (White Paper Presentation by Fouad Tobagi)

Multimedia applications place new requirements on networks and their protocols, data rates, traffic patterns, loss, latency, modes of communication, synchronization, etc. Multimedia traffic characteristics differ substantially from those of more traditional data traffic. Existing networks and protocols are not capable of satisfying these new requirements. Thus, new network infrastructures and protocols are being developed.

Ethernet is the most commonly used Local Area Network (LAN) scheme. It uses Carrier Sense Multiple Access with Collision Detection (CSMA/CD) to allow multiple stations to share a single channel. This channel has a bandwidth of 10Mb/s. A maximum cable length of 100m limits the maximum station-to-station distance to 200m. The Institute of Electrical and Electronic Engineers (IEEE) is currently working on a 100Mb/s Ethernet standard. The other LAN scheme in use today is the token-passing ring: Stations are attached to a ring network in which a token is circulated to control access to the ring. One type of token-passing ring, known as Fibre Distributed Data Interface (FDDI), has a bandwidth of 100Mb/s and support for synchronous traffic.

Asynchronous Transfer Mode (ATM) has emerged as the most suitable switching scheme to handle Broadband Integrated Services Digital Network (BISDN) traffic. The ATM architecture consists of the ATM physical layer, the ATM layer, and the ATM adaptation layer. The physical layer is an interface to the transmission medium. The ATM layer is responsible for providing cell transport and congestion control. A cell is a fixed size (53 bytes) protocol data unit (PDU). The ATM adaptation layer (AAL) protocols provide functions to the higher layers that are specific to the type of service required. ATM signaling is based on a set of messages that are used for dynamically establishing, maintaining, and clearing ATM connections.

To provide service for the emerging multimedia applications, there is a need for (i) new routing algorithms which are able to take into account the requirements of bandwidth, latency, and multipoint communications when finding routes; (ii) new routing protocols with support for streaming (virtual-circuit-like) capabilities, resource reservation, and multicasting; and (iii) new higher-capacity routers, with support for integrated services. The older Internet routing protocol uses the Bellman-Ford algorithm, whereas the new protocol uses Dijkstra's algorithm. The link label assignment scheme is more flexible and the convergence is faster in the new protocol. Active research is also going on in resource reservation protocols, which are responsible for allocating network resources for multimedia traffic. They are, to a large extent, independent of the routing protocols. The two most important resource reservation protocols are the Internet Stream Protocol version 2 (ST-II) and the Resource ReSerVation Protocol

As with the network layer discussed above, the transport layer needs new protocols suitable for multimedia applications. These protocols should be efficiently implemented and provide timing information, semi-reliability, multicasting, error recovery mechanisms, and rate control. The emerging transport protocols suitable for multimedia are the Xpress Transport Protocol (XTP) and Real-Time Transport Protocol (RTP). Session layer protocols are also being actively developed. There is a growing interest in supporting digital video applications over local area networks. Since, video traffic characteristics differ substantially from those of data applications, new servers capable of handling the specific characteristics of video files and traffic are needed. Applications will require storage capacities that are one or two orders of magnitude larger than what is presently available. The options for providing this increased storage are (i) increase the number of disks or (ii) use tertiary storage such as optical jukeboxes or robotiocally manipulated tape libraries. The former is limited, while the latter is open to research.

5.2 Position Statements

5.2.1 Derek McAuley

In circuit-switched networks, quality of service is trivially guaranteed. But, in packet switched networks, the situation is quite the opposite. The problem is that packet-switched networks have far less resources than is required to satisfy the peak demands of all the customers simultaneously. There are two approaches to the problem. One suggests that guarantees can be provided as long as the application declares its requirements in advance. The guarantees are statistical, however. The other suggests that the network shouldn't provide much more than best-effort; applications can be made to adapt intelligently using the increased processing power available on every desktop. The latter is prone to instability and a reliance on the end-systems playing the same game.

While the network is free to modify its service, it should implement this in a way that it provides applications with information they require to adapt:

- Networks should provide a service which is to guarantee not to modify the inband part of communication more frequently than some agreed rate; hence applications know that any computation they perform to adapt their behavior is cost-effective.
- Networks should inform end-systems (applications) of modifications to their resources in a timely manner to enable them to take appropriate action, rather than having to rely on observational data (like TCP).

5.2.2 Stephen Pink

The goal of multimedia networking is to integrate real-time voice and video applications into the distributed system platform. The question is whether and how the Internet will have to change. An important constraint in answering this question is that the network will have to work for traditional applications just as well.

Internet is rapidly expanding and Multimedia applications are becoming popular. There are two schools of thought for accommodating the resulting heavy flow of traffic: Change the service interface or change the queueing disciplines in the gateways. The latter appears to be easier to accomplish since the interface need not be re-written. However, the former may not be hard if the change occurs soon and only to delay-constrained applications. This is so because there are not many multimedia applications at present. Those in the former school have introduced what is called a flowspec, which the user passes to the network for characterizing the resources needed by the application. This could be used for establishing a flow with an associated quality of service. A flow is something between a virtual circuit and a datagram: Although there is no end-to-end connection, there is a temporary path established upon which datagrams are switched. If the path times out, it will be refreshed by new path messages that work

Protocols that exist today require path establishment and resource reservation to be made simultaneously. This makes network scheduling very inflexible. For example, if a group of people desires to teleconference at a predetermined time in the future, it is impossible to reserve resources in advance. In the present scheme, pre-knowledge cannot be capitalised on. The problem in permitting advance reservations is that it expands the state that the network must maintain into a third dimension: time. Due to the resulting complexity, it might be better to abandon that model in favor of one that will scale better, yet offer the same service. One approach to alleviate this problem would be to carry out call admission based on a statistical view of the traffic in the network. If this works, then the network itself would not need to keep nearly as much state as on the connection-oriented model, since knowledge of present and future use of the network would only need to be on users' workstations.

5.2.3 Stefan Covaci

Optical technology offers practically infinite bandwidth at very low error probability. Thus, a series of performance bottlenecks are becoming increasingly more important. In order to overcome some of these bottlenecks, a system architecture was proposed based on the view of the communication network as a bus on which the processing units of end-systems are attached. Further, this bus is viewed as a huge storage device, since optical technology provides high bandwidth, low attenuation links: A link with bandwidth R and transmission delay T has a storage capacity of RT. The write time for this memory is the media access latency. The virtually unlimited storage capacity should bring down the write time to almost zero. The read time is essentially determined by the distance between the read tap and the packet underway in the network, and is limited by the speed of light. Two scenarios are envisaged, one in which all the processing units are attached to one shared memory, and one in which a multitude of shared

The proposed system architecture alleviates a number of bottlenecks, some of which are listed below:

- Eliminates the need for complicated procedures to guarantee end-to-end QOS, since the system provides a single service satisfying the highest QOS required. (e.g., no need for several AALs)
- Eliminates the host-network interface (HNI) bottleneck since there is only one communication system - the HNI is now an I/O device for the share-memory.
- Provides high storage capacity and I/O bandwidth for continuous media applications.
- Permits the design of a new flexible operating system (OS) that is relieved from the overhead related to pure communication and of the task of arbitrating accesses to limited resources (I/O bandwidth, memory).

We summarize below some of the network-related issues that were put forward:

- Jitter for isochronous services should not exceed the time between two successive memory-write operations requested by a single user.
- Since access times are very small, the network should provide users with connectionless service.
- The network must provide concurrent access to multiple users in order to behave as a shared, multiported memory.
- Error protection: one mechanism since one service.
- Fixed-length/variable-length cells: needs further investigation.
- The network aiming to be global must implement a synchronous digital hierarchy.
- Addressing schemes will have to accommodate a very large number of users, e.g., E.164.

5.2.4 Aurel Lazar

In this position paper, an architecture for multimedia networks is proposed and related to the integrated reference model (IRM), a model for the organization of information transport entities, network entities and operators on such entities in broadband networks. From the logical standpoint, a multimedia network can be viewed as set of three planes which form roughly a three-level hierarchy. In this hierarchy, the underlying broadband network and media processors lie on the bottom plane. The multimedia network (middleware) lies on the middle plane. The services and applications lie one the top plane. The interface between the bottom and middle planes provides quality of service (QOS) abstractions, while the interface between the middle and top planes provides service abstractions. The functionalities of each plane was shown to fit into the mold of the Extended Reference Model (XRM), an extension of the IRM to multimedia networks. The multimedia networking architecture proposed above follows the client-serve

The main concept underlying QOS abstractions is that of the schedulable region of a multiplexer. It is defined as the set of points in the space of possible calls for which QOS can be guaranteed at the cell-level. It is a stability concept. From the point of admission control, the schedulable region is a complete representation of a link. The concept can be applied to any scheduling algorithm. The other concept discussed in relation to QOS abstractions is that of the multimedia capacity region. The set of combinations of calls for which QOS guarantees can be provided at the fame level is called the multimedia capacity region of the audio-video/data-storage unit in a customer premises equipment (CPE). It abstracts away the lower level details like the operating system and protocol processing overheads.

For binding services with resources, an open architecture is proposed: The network entities being bound are modeled as communicating objects with well-defined interfaces that can be invoked externally by binding algorithms. A set of well-defined methods and global primitives is used for this invocation. All interfaces are defined using the Common Request Broker Architecture (CORBA) Interface Definition Language (IDL). All instantiations of interfaces reside in a repository called the Binding Interface Base (BIB), which provides multimedia networking abstractions for producers, consumers and media processors. CORBA is used for communication among objects. The above architecture supports any proprietary binding algorithm. The network management architecture is designed around the basic manager agent interaction. Information on managed resources is stored in repositories called Management Interface Bases (MIB).

From the physical standpoint, the following network abstractions were introduced: Switches are considered random access memories while communication links are considered first-in-first-out (FIFO) memories. Thus, the network is a global distributed memory in which communication takes the form of a series of reads and writes. Conceptually, the entire communication network is identical to a workstation.

5.3 Research Items

The following issues were discussed as questions for future research on multimedia networking and communication:

- The main issue in the design of a 100Mb/s Ethernet is to devise physical layer protocols that can operate at such high bandwidths.
- Even though existing FDDI chips support both synchronous and asynchronous modes, currently only the asynchronous mode is used due to lack of a well-defined bandwidth allocation procedure.
- To provide for the emerging multimedia applications, there is a need for (i) new routing algorithms which are able to take into account the requirements of bandwidth, latency, and multipoint communications when finding routes; (ii) new routing protocols with support for streaming (virtual-circuit-like) capabilities, resource reservation, and multicasting; and (iii) new higher-capacity routers, with support for integrated services.
- How can routing protocols take advantage of any knowledge of the traffic on the links?
- How should file servers be designed to handle video files and traffic?
- More work is required on adaptive coding schemes and environments in which it is possible to write adaptive applications. This requires applications to know not just how much network capacity is available, but also bus bandwidth, processor cycles, memory, etc.
- The network control mechanisms need to be more open in providing information to end-systems and applications on what is really going on. Using observations to deduce this state is inefficient and often the availability of information is delayed.
- When considering personal end-systems, mechanisms need to be in place to enable the users to overlay their policies on the applications given what the network is saying it can provide.
- How must the Internet change to support integrated services? There are two schools of thought for accommodating the heavy flow of traffic: Change the service interface or change the queueing disciplines in the gateways.
- How can the network offer advance resource reservations without making the system too complex?
- Can call admission be based on a statistical view of the network traffic? If so, then the network itself would not need to keep nearly as much state as on the connection-oriented model, since knowledge of present and future use of the network would only be on users' workstations.
- In a network-memory, transmission delay and storage capacity are inversely related. How do we overcome this difficulty?
- New operating systems have to be devised for supporting session-oriented distributed processing.

5 Multimedia Documents and Mailing

Thomas Meyer-Boudnik, University of Mannheim, Germany
Blair MacIntyre, Columbia University, USA

5.1 Survey (White Paper Presentation by Gerd Schürmann)

Electronic mail is widely used as a means of asynchronous communication between computer users. However, message content is typically simple text. More complex structure and content - multimedia messages - are currently limited to isolated communities.

In this context the term “multimedia” is associated with the combination of different information entities which are intended for human perception. Multimedia messages may be composed of information entities such as character text, graphics, moving and still images, audio, interleaved moving image and audio streams, and compound document. Furthermore, link structures can be imposed on a message. The link structure may be used for annotation purposes, for example, and can result in a presentation order of the message components which differs from their sequence within the message.

Various multimedia-mail systems have been around for over 10 years, each supporting its own proprietary multimedia-message format. Unfortunately, the de facto standard for Internet messages is text-only; only recently have multimedia extensions been proposed and implementations begun to emerge. With the growing acceptance of the two competing standards – the CCITT X.400 (‘88) series of recommendations and the Internet MIME proposal – incompatibilities between the many proprietary electronic mail systems are no longer a major issue. Interoperability will be possible in the near future even though only text (i.e., the International Alphabet No. 5) is commonly supported and used.

Additionally, multimedia-mail systems which conform to the standard can be used as a basis for various other services, such as asynchronous directory access, and can be considered the basic components for group communication.

A variety of electronic mail prototypes which supported the inclusion of images and audio in addition to text, for example, were developed in the eighties, including the DARPA experimental Multimedia Mail System, the Distributed Interoffice

Mail System and Diamond Mail from BBN. Currently, many mail tools support editing and viewing of multimedia message contents based on proprietary formats. For global message interchange, gateways are provided, for example, to Internet SMTP, requiring a bilateral agreement between messaging parties on the necessary conversion from the proprietary format into SMTP simple text messages.

The challenge of extending text-based messaging, such as Internet SMTP, to multimedia messaging has been addressed by the Internet Engineering Task Force (IETF) Working Group with the Internet MIME (Multipurpose Internet Mail Extension) message structure. It supports multimedia message content as well as references to externally-stored content parts. An alternative approach is the development of the Multimedia-Mail Teleservice based on CCITT Recommendation X.400(88), currently under development within the BERKOM project funded by the German PTT. In addition to defining a standard message structure, X.400 (and, to a limited extent, MIME) attempt to alleviate the problems created within the message transfer system by large messages, such as those with video content, by complementing the store-and-forward mechanism inherent to electronic mail with additional exchange mechanisms.

Two other projects are also developing multimedia-mail systems similar to the BERKOM system: the RACE project R2060 (Coordination, Implementation and Operation of Multimedia Teleservices (CIO)), and the RACE project R2008 (Euro-Bridge).

Extension of Internet Mail: MIME

MIME offers a simple standardized way to represent and encode a wide variety of media types, including textual data in non-ASCII character sets, for transmission via Internet mail. To allow for the graceful evolution of Internet mail facilities, MIME limits mail bodies to 7-bit ASCII text and line-oriented data of bounded line lengths. To permit the continued evolution of mail facilities to an ever-expanding set of data types, MIME introduces a flexible two-level mechanism for naming data types, and a simple procedure whereby new types can be registered with Internet authorities.

A complete description of MIME is beyond the scope of this report. The MIME standard defines seven primary content types, including four straightforward types (text, image, audio and video), a message type for encapsulation of other messages, a multipart type and an application type. The multipart type allows multiple types of data within a single message, both as separate logical components and alternative representations of the same logical component. The application type is for data that does not fit within the other categories. Each primary content type supports multiple subtypes, with the expectation that new innovations and extensions will take place via the definition of new subtypes.

ITU-X.400 and the BERKOM-Multimedia Mail Teleservice

The BERKOM-Multimedia Mail Teleservice being developed within the BERKOM project (BERliner KOMmunikationssystem). The BERKOM-Multimedia Mail Teleservice (BMMMTS) is based on the principles of the 1988 version of the CCITT Recommendation X.400 Message Handling System. The X.400 Message Transfer System (MTS) delivers the messages submitted to it by either a User Agent (UA) or a Message Store (MS) to one or more recipient UAs or MSs, and can return notifications to the originator UA. All mandatory service elements available to the user of an X.400(88) system are available to the user of the BERKOM Multimedia-Mail Teleservice.

The interpersonal messaging (IPM) service has been enhanced in order to provide for additional capabilities to include multimedia information within a X.400 message. Separate components for the handling of external references to information which cannot be directly included in a message are provided, including both global and local stores for the external data. Comprehensive handling of external references is perhaps the main advantage of this system over MIME for multimedia mail. Since typical multimedia messages might be too large for message transport systems to handle, as well as potentially exceeding the storage capacities at both intermediate and the recipients' sites, the ability to pass references to data stored in globally accessible data stores is extremely important. A common alternative is to split such a message into multiple parts and deliver them separately. However, keeping track of these multiple messages in order to reconstruct the original message is outside the scope of X.400 and, moreover, does not solve the principle problem. In contrast to messages transferred as one unit, this deferred transfer of message content requires the specification of strategies for message access, in particular when a message component, which may be referenced by more than one recipient, shall be deleted. The Global Store Server (GSS) offers a chargeable storage service to make any data, especially high volume data, accessible world-wide. It can be considered as a public or private value-added-service for temporary deposition of bulk data in a global network.

The obvious question is "Who will manage the Global Store?" When a commercial organization provides the service, what happens with "old messages?" How long do they stay on-line? One solution is for such information to be encapsulated in the object reference, allowing the sender to control the time-fidelity of the object storage, and hopefully ensuring that the receiver is aware of the life-span of the object reference. Furthermore, anybody with network access can provide a "global store," so cheap, long-term storage is not a problem.

Interworking Between MIME and X.400

Interworking between X.400(88) and MIME is well defined in various standards documents, so systems based on either standard can communicate with each other. The major differences between the two approaches, besides the more political distinction that X.400 is a formal International standard within ISO and ITU and MIME is an Internet standard, are in the support of many important header attributes in X.400, such as support for “confirmation” of messages. However, this functionality is not multimedia specific and, as was pointed out during the discussion, it is not clear how useful many of these additional features really are. For example, it is impossible to verify that a user has actually read the contents of a message, regardless of whether or not it is delivered.

MIME and the Multimedia-Mail Teleservice based on X.400(88) provide reasonable support for multimedia messages. The later provides a more comprehensive solution to the problem of large message contents inherent to multimedia mail by an additional exchange mechanism allowing the resolution of references to externally stored message content.

Multimedia-mail can serve as the basis for asynchronous distributed applications. Perhaps the most promising application area is CSCW or Groupware. This includes work flow automation, which encompasses information routing, task automation, and decision support. One leading category of messaging-centric applications in this area is group scheduling and calendaring, which supports the planning of meetings and allocation of resources, such as conference rooms and equipment.

Security issues, such as confidentiality, integrity, authentication, access control, non-reputiation, audit and key-management, are among the most important issues to be solved in the near future for multimedia electronic mail. However, these issues should probably be addressed by cryptographers and multimedia researcher should focus on multimedia specific issues.

5.2 Position Statements

5.2.1 Simon Gibbs

This talk focused on documents in general, not just mail, by examining how documents can be composed. Since the essence of multimedia lies in composing a structure between elements from diverse media, to understand multimedia, we must first understand composition. In particular, the speaker discussed extending element types to include “live” data.

What is the difference between stored and live data? With stored data the sink is “in control” – it has the choice of selecting what data to receive and when to receive it. With live data the situation is reversed. The source is “in control” and the sinks have little choice in what is sent their way. Even though they may not sound particularly flexible, there are many situations when live data sources may be more efficient or more timely than stored data, such as news wires and live video feeds.

Composition is the essential task for authors of multimedia documents. Several generic composition mechanisms have been identified:

- *spatial composition*: the positioning of media elements in 2D or 3D space.
- *temporal composition*: positioning time-based media elements along a temporal axis.
- *semantic composition*: explicit links and other semantic relationships between related material.
- *procedural composition*: express associations between media procedurally.

Although identified above as four distinct mechanisms, in practice they are often mixed. One can expect "rich" document models and authoring environments to support most, if not all, of the above mechanisms. In particular, current standards activities (MHEG, HyTime, HyperODA) and commercial activities (e.g., ScriptX) combine several composition mechanisms.

Live data can also be included in the composition model: aside from choosing whether or not to ignore a data stream, applications can also *filter*, or process, the stream. A live data stream has three basic *components*: sources, sinks, and filters. Connecting filter and sink components allow multimedia documents to be constructed which select and display live data. In addition to the capabilities of equivalent static components, live data components continually process and display the incoming data streams. Using them, we can create documents that can be "patched in" to new network services. These new services, incorporating broadcasts and multicasts from live data sources, are emerging as bandwidth increases and protocols evolve.

Based on the discussion, it is not clear how useful the concept of live data is. For example, the idea of embedding live data objects can be thought of as an instance of the object/application embedding idea. If we can have multiple application-objects embedded in a document, live-data objects can be thought of as simply the output of an application that receives and processes a data stream. Of course, allowing such objects to be embedded in documents raises issues of synchronization between the cooperating applications that are not as critical with typical embedded applications, which are only activated as a result of user action.

5.2.2 Erich Neuhold

This position statement discussed the framework for a distributed multimedia archiving system that will be needed to support multimedia hyperlinked documents and both synchronous and asynchronous cooperation via high speed networks.

Most multimedia applications involve a diversity of conventional data types like numbers, text, and tables combined with media data like images, graphics, audio, video and animations. An important difference between multimedia and traditional databases is that users should be able to control presentation of continuous media to allow for more than conventional linear consumption, such as controlling the rate of video playback. Furthermore, each data element could be represented using different formats, such as different audio and image formats for the same sound or picture data.

The variety of datatypes in a multimedia database also imposes new requirements on consistency checking, indexing and searching. Documents must be classified somehow to facilitate these activities. One approach to maintaining document consistency across a large database is to define an SGML document-type definition (DTD) whose instances describe documents, essentially a super-DTD or meta-DTD, and requiring all documents in the database to have an associated DTD. By using an SGML super-DTD, much of the semantic information required to access the data can be embedded in the documents. Of course, a document description standard will never encompass all documents, so the super-DTD must be flexible enough to define new or non-standard types.

The speaker discussed an archiving and retrieval teleservice for multimedia documents, called Multimedia Archiving (MMA), using multimedia mail as a means for interchanging multimedia documents between archive clients and an active multimedia archive server. Using mail as the access mechanism solves a variety of problems: only X.400 documents need be supported (gateways can handle the conversion from other types), and the various document transfer problems are already addressed by the mail transfer system.

Descriptive search criteria can be used to search for documents by addressing document contents as well as multimedia specific data. For example, this allows documents to be selected which do not contain video clips longer than 1 minute. Another important feature is support for dynamic document composition by the archive. This allows retrieved documents to be dynamically created that conform to the users requirements, like having no video or having images represented in a certain format. Other queries, such a returning only the document description or the number of query matches, are supported. A sample application, the Calendar of Events (CoE) was discussed.

Multimedia databases will benefit from database management systems (DBMSs) supporting general-purpose schemas which can model the complex semantics of typed hypermedia objects, by freeing applications from reimplementing these semantics. Object-oriented DBMSs are particularly well suited to capturing these semantics. The concepts for time-dependency and synchronized presentation of multimedia data must be integrated in the data description and query languages. Furthermore, presentations and control of presentations at the user's workstation requires a client server architecture, specific buffering concepts, and networks supporting continuous or isochronous transport protocols.

The discussion turned to the requirements of electronic publishing of multimedia documents. Specifically, how can we ensure high quality documents, verify the accuracy of an electronic document and trace electronic documents. In contrast, nothing in the current storehouse of multimedia documents, the World Wide Web (WWW), can be trusted.

Various approaches were discussed, but it would seem that the job of the multimedia community should be to facilitate the creation of high quality documents and to encourage cryptographers to develop ways of verifying and tracing documents.

5.3 Research Items

One of the major issues brought up during the discussion is that, contrary to the speakers assertion that the major issues of multimedia mail have essentially been solved, very few of us use multimedia mail. If not even the multimedia researchers are using it, how can we assert that the problem is solved? The major problem seems to be that the majority of mail composition and reading tools used by researchers in the Unix world are still text-based. Thus, even if I have good tools for composing multimedia mail, it is likely the recipients of my mail will read it in text form. Ironically, limited multimedia capabilities are far more common among the business (PC/Macintosh) community, where proprietary graphical mail proprietary graphical mail programs are common. In the following the resulting research questions have been classified into the areas of real-time support for continuous media, tools for interactive creation of multimedia messages, and using multimedia documents to facilitate more powerful applications.

- How can BERKOM-type Global Stores be extended to support controlled real-time retrieval of data elements such as video?
- Tools for composition and viewing of multimedia messages need to be created. Where should the specification of dynamic document composition operations required by the user come from, and how should it be performed by the user? Is an interactive, graphical specification an adequate approach and, if so, what form should it take?
- Multimedia mailing bears some potential for multimedia enhanced work flow management. Audio and video annotations, for example, can be used to add some kind of informal interaction between participating users to todays work-flow management paradigms which mostly do only support formal or semi-formal interaction. Can an adequate multimedia enhanced work flow management model be built?
- To support the exchange of multimedia documents between authoring systems and multimedia mail, two approaches are possible: (1) mapping of standardized document formats into a mail-internal document format, or (2) explicit support of standardized document formats by mail. Which if these approaches should be used?

6 Conferencing and Collaborative Computing

Michael Altenhofen, Digital Equipment, Germany

6.1 Introduction (White Paper Presentation by Eve M. Schooler)

Definitions and Taxonomy

Collaborative computing “encompasses the application of computers for coordination and cooperation of two or more people who attempt to perform a problem together”. The collaboration matrix spans across which can be used to categorize cooperative, or *groupware* systems:

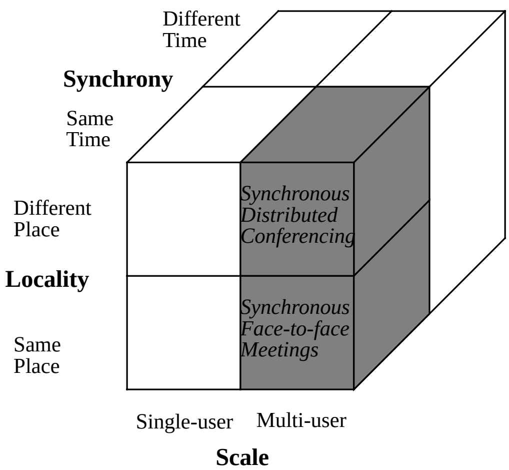

Figure 1: Collaboration Matrix.

The most notable dimension is *time*. Cooperation might take place at the same time, i.e. *synchronously*, as with computer-supported meetings, or at different times, i.e. *asynchronously,* as with electronic mail systems. A second criteria is *locality.* Are groups that cooperate via computers co-located (in one room, using a liveboard) or geographically distributed? The third axis ration space according to *scalability*. Here, the main question is how well systems scale up to support a growing number of users.

Conferencing System Components

This session focuses on *conferencing*, which is one form of synchronous tele-collaboration. Conferencing systems usually combine shared computer-based workspaces with real-time communication channels, such as video and/or audio.

Shared workspaces allow group members to jointly view or manipulate data displayed by one or more computer applications while maintaining data consistency. Data manipulation is controlled by *floor policies*. Different floor policies are achievable depending on the level of simultaneity (the number of active users allowed), the granularity at which to enforce access control, e.g., whole documents vs. single paragraphs), and the way the floor is passed among users.

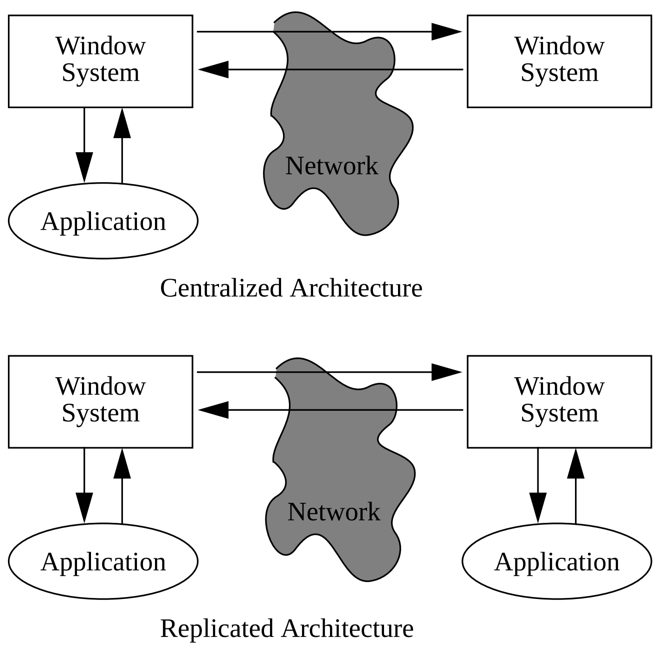

Figure 2: Shared Workspace Architectures

Three different architectures have been deployed to implement shared workspaces.: *centralized, replicated,* and *hybrid*. In a centralized model, applications only run at one site. Input from the floor holder is passed back to this site and the views are synchronized by broadcasting all output to all conference sites. Within this scheme, existing single-user applications can be transparently turned into groupware applications. In a fully replicated architecture, each site runs its own copy of the application. Here, the input is broadcast to all sites and the views are then updated locally. This normally requires specialized, collaboration-aware applications, but yields better performance in WAN scenarios. Hybrid approaches, in turn, mix both approaches by combining a centralized data repository with replicated graphical front-ends.

Audio/video data streams are used to supplement shared workspaces with additional communication channels and conversational cues found in traditional face-to-face meetings. Whereas earlier systems coupled analog audio/video transmission with computer based workspaces, there is a trend to fully integrate these media types into digital computer systems. Then, audio and video streams can even be considered as part of the data shared in the conference workspace.

Session Papers

The following sections summarize the presentations and discussions of the "Conferencing and Collaborative Computing" session. The whitepaper by Eve Schooler contains a number of architectural considerations that can help to enable wide spread telecollaboration. Henning Schulzrinne's talk analyses problems in various areas that result from the fact of conferencing being a vertical application. The last presentation by Max Mühlhäuser and Tom Rüdebusch presents a software technology for the development of customized conferencing/groupware solutions.

6.2 Issues on Widespread Telecollaboration

If widespread telecollaboration shall become reality, interoperable solutions will have to be found. Interoperable solutions based on standards will simplify the development process for collaborative systems by providing common, re-usable components. Furthermore, interoperability, through shared abstractions and standard interfaces, will help to master heterogeneity that will facilitate widespread usage.

Communication Underpinnings

Synchronous telecollaboration often involves tight interaction among a (potentially large) number of individuals through different types of media that have varied characteristics. Interactiveness can be affected by communication delays, either update delays in shared workspaces or end-to-end delays in real-time media. Thus, powerful communication services, i.e., standardized protocol suites, are needed that are able to transit data in real-time with minimal delay using group-modes of communication.

Another issue in this context is scale; in a unicast distribution scheme bandwidth requirements are prohibitive for large groups, so multicast support is fairly essential for efficient data transport. Yet, mechanisms have to be devised that address the problems with group address management.

Efficient distribution is also bound to the availability of network resources. Resource management and quality of service (QoS) negotiations are the key concepts here, but the emerging idea is that the network should be able to signal changes to applications and applications should be able to adapt to new situations.

Architectural Models for Widespread Collaborations

Several attempts have been made to develop abstract models for conferencing systems. They typically have tried to introduce a common taxonomy, or to partition system functionality, or to identify information flow, or to specify component interfaces.

The simplified model that is depicted in Figure 3 is based on the principle that, despite the different requirements and usage patterns, media control can be separated from media transport.

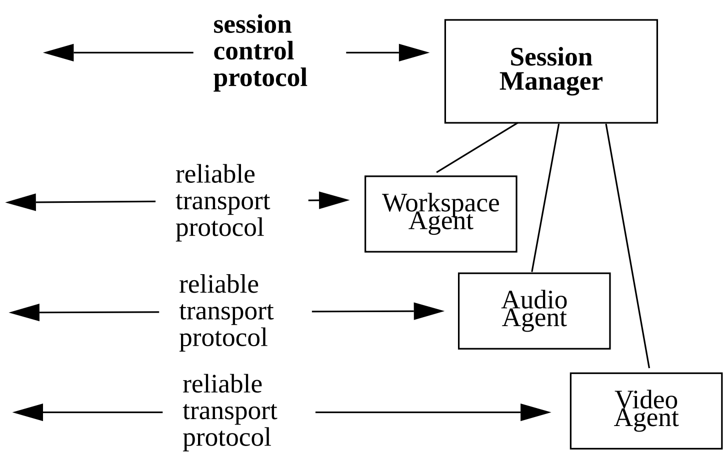

Figure 3: Session Control Architecture

The “heart” of this distributed architecture is a re-usable session manager that is decoupled from both the application and the underlying media agents. This separation serves two purposes: First, it provides a generic control layer that conferencing tools can build on without duplication of effort. Second, it promotes the development of replaceable media agents that can be plugged in to accommodate the diversity in hardware capabilities and user preferences. Session managers are also the centre of control flow, both locally and remotely. At one site, they mediate information exchange among the media agents; inter-site communication happens among peer session managers.

Collaboration Policies

Although the collaboration model introduced above combines all media under a uniform control scheme, a comprehensive support of conferencing scenarios has to take their different control needs into account. Sessions are not only characterized by their members and the media that are used, but also by the set of policies that rule the interactions among them (e.g. who may join a session, when and how a session may be modified, etc.). Flexibility should be supported in different ways.

Policies can be implemented in replaceable modules that are loaded and selected at session run-time. An alternative approach is through policy-based control protocols that rely on a common session substrate for multiparty agreement but which implement different policies via a specification language.

Control Models and Mechanisms

Once sessions have been established and appropriate collaboration policies have been chosen, some level of coordination among participants has to be guaranteed. This is done by disseminating (parts of) the session control state to the session managers at the participating sites.

Control models are differentiated by whether control is centralized in a separate component or truly distributed, and whether state consistency is always guaranteed through reliable synchronous messaging or whether it is eventually reached through periodic refreshes.

The latter approach, known as *light-weight sessions*, has become quite successful in the Internet through the Multicast Backbone (MBone) tools. Here, control is completely decentralized (without explicit coordination) with each site multicasting its own state to other parties. This scheme is quite feasible for large sessions with loose control, yet further investigations are necessary to find out how this approach maps to scenarios where tighter control is essential.

Distributed Messaging

Tighter control is especially needed in the area of shared workspaces where, at some point in time, participants need to be sure that their views are virtually identical. This requires stricter multiway distribution mechanisms to assure global synchrony of shared state.

Traditional inter-object communication mechanisms, like Remote Procedure Call (RPC), do not match very well since they assume a client-server relationship between the communicating entities. Multimedia collaboration often follows a peer-to-peer communication paradigm with a strong emphasis on group-oriented dissemination.

Implications of Heterogeneity

Computer systems that are in use today are diverse and will remain so for quite a while. Thus, widespread telecollaboration will heavily depend on architectures that can cope with the varied capabilities of the end systems. There must be ways to describe and characterize the capabilities and requirements (e.g., through self-describing media agents) and to export these specifications (e.g., through configuration resource directories) so that interoperable solutions can be found.

Even with a negotiation scheme problems remain if no consensus can be reached, e.g., if peers do not support a common media encoding format. A general solution to this problem has been suggested by means of so-called *combination nodes*. Combination nodes are hardware or software modules, either deployed in end systems or in the network, that allow media streams to be combined, translated, mixed, or selected as they flow from senders to receivers. Obviously, such nodes could also help to further reduce network bandwidth requirements.

Synchronization

In the context of multiparty, multimedia collaboration synchronization issues appear at various levels.

First, synchronization of different media is necessary to convey the semantic relationship of different activities (e.g., audio and workspace activity). This synchronization can easily be achieved by bundling the different media during transport. However, from a heterogeneity point-of-view, synchronization of media streams, through timestamps or adaptive techniques, seem to be more appropriate for a number of reasons: First, bandwidth and QoS requirements are easier to handle if the inherently different media are treated as separate streams. Second, separate streams provide more flexibility in that users can opt to receive different combinations of media streams.

Another place where synchronization is required is inter-site coordination. In other words, to share a global workspace state events might have to be delivered simultaneously to all sites.

Floor Control

A third form of synchronization in conferencing systems is introduced through coordinated access to shared information. Such floor control is fairly essential if the number of participants in sessions becomes large. Within a unified conferencing architecture different floor control policies are conceivable.

One scenario may require separate floor control for the different media. In shared workspaces, floor control is mainly used to guarantee data consistency, or to establish a social protocol. Yet, data consistency is of less or no concern for real-time audio and video. Here, floor policies may be introduced to reduce bandwidth consumption.

In an integrated approach, floor control can apply to multiple media to allow policies that reflect the group context, e.g., video-to-follow-audio, or video-to-follow-workspace activity.

Rendezvous

Another problem that needs to be addressed for widespread teleconferencing is the question of how to find users and conferences. Methods that have been devised fall in two categories:

Synchronous methods are based on directory services that keep track of and announce conferences using multicast. This is well suited for public sessions (of large scale) that can be joined by anyone. Conferences that are limited in scope or that are of private nature are better supported through explicit invitations. Problems remain, though, in the area of user location, so better address schemes have to be developed.

Asynchronous schemes make use of existing tools and infrastructure. The most prominent examples are electronic mail, where active-mail extensions can be deployed for group session establishment, and the WorldWide Web (WWW) where work is underway to provide synchronous rendezs on documents that appear as pages in the web.

6.3 Conferencing as a Vertical Application

As mentioned earlier, conferencing systems try to allow geographically distributed users to virtually meet and work together as if they were in one place. How well this illusion works out - and as a consequence, how well such systems are accepted by end users - depends on the appropriate support from the underlying components and services.

Media Quality

Current systems largely fail to imitate physical conferencing situations because of the poor communication media quality. There are several, not necessarily technical reasons for this, like bad or wrong equipment (low-resolution cameras, microphone/speaker combinations that are unable to deal with acoustic feedback), adverse environments (noisy offices with inappropriate lighting), or simply limited system resources (CPU power, network bandwidth, screen real-estate).

One way to cope with these problems is to prioritize communication channels. For instance, both practical experiences and formal experiments show that in many cases people prefer good audio quality over good video quality, i.e., frame losses are much more acceptable than audio drop-outs.

Spatial cues could also be used to improve communication media quality. They could either be real, like indicating the location of a speaker in a room, or artificial, like "placing" people around a virtual conference desk. This can lead to scenarios that even go beyond traditional physical conferencing situations.

Networking Issues

Networks as they are deployed today are not very well suited for flexible computer-based conferencing. Furthermore, it's rather questionable whether proposed catch-all technologies like ATM (Asynchronous Transfer Mode) or protocols like XTP (Express Transport Protocol) can solve all outstanding problems in a satisfactory way.

An alternative approach to a "perfect" network is to make applications "elastic" in a way that they can adapt to variations in the service quality provided by "not-so-perfect" network. In such an environment a possible service model could be:

The network provides two types of services, a guaranteed constant bit-rate (CBR) and an available bit-rate service (ABR) service. Applications use the CBR service to allocate the minimum bandwidth they need to work in an acceptable way. Bandwidth that is needed to enhance quality is acquired via ABR channels.

As a variation of this overall scheme the network could signal when more CBR bandwidth is available, giving applications a chance to acquire, for a certain period in time, more guaranteed bandwidth. This allocation scheme could be coupled with a pricing scheme yielding different types of service classes (e.g., "teenager" services with degradable quality or "executive" services with constant, high quality). The question remains how many service classes are feasible in terms of manage ability and billing schemes.

Operating System Support

The problems in the area of operating system support for real-time multimedia conferencing are twofold. On the one hand, the scheduling policies deployed both in standard multitasking and real-time operating systems do not meet the needs of conferencing systems.

The second problem lies in the question of how far (multi)media or even conferencing services should be embedded in the operating system. Media tools are fairly complex, so, from a portability point-of-view, cross-platform APIs (Application Programming Interfaces) could help to minimize programming time and effort.

Generic operating system support would also prevent application programmers from re-inventing the wheel, but given the diversity of media usage and manipulation in applications, such an API would probably tend to be fairly “wide” (in terms of functions and parameters) and that would make it rather unusable.

Conferencing Frameworks

Existing conferencing systems often seem to concentrate on a limited set of scenarios where the functionality is adjusted to these scenarios and is only accessible as part of a monolithic block.

As a recent trend, new systems attempt to explore the richness of the human communication/cooperation patterns that pass beyond traditional small-scale group meeting scenarios. Examples are unplanned hallway encounters, drop-in seminars, panel discussions, and jury trials. These different communication/cooperation patterns obviously cannot be modeled with a single scheme. They require systems that can be combined in a flexible way.

The approach proposed here follows the traditional Unix filter paradigm: Generic reusable tools with well-defined (simple) functionality act as building blocks that can be combined or connected via a simple mechanism, the so-called pipe. the tools process information, they don’t know where it comes from and the don’t care to whom the processed information will be passed.

In contrast to this pipe model that implements a strict sequential flow of information, the connectivity between tools in this approach is based on the model of “anonymous message passing”: Tools export their functionality and (parts of) their internal state to the outside world through a message interface. Other parties can remote control such a tool or declare interest of changes in its internal state.

One possible implementation is based on an *application-level multicast* where a central component, the message replicator acts as a local message dispatcher.

Two major benefits arise from this scheme: First, media agents (which are often expensive to implement) can be reused and tied together in different ways by different control agents depending on the scale and the pattern of the conferencing scenario. Second, it’s fairly easy to add new components that combine information from other sources in new ways, like statistics or logging tools.

However, more work needs to be done to explore implications of this scheme on central resources (like floor control), error reporting, and security.

6.4 Context Embedding and Reuse in Cooperative-Software Development

Although conferencing and collaborative computing are established concepts and have been in use for quite some time now, systems have often fallen short of expectations as effective means of communication and cooperation. A number of shortcomings in state-of-the-art desktop videoconferencing and groupware systems need to be addressed in future cooperative-software development.

Context Embedding and Customization

Current conferencing and groupware systems mostly strive for generic, service-type solutions, thus neglecting the operational and organizational context in which they are used.

They tend to turn the world upside-down by demanding that the group or social activities have to take place in the context of the conferencing system rather than embedding the conferencing technology into the application domain. Future systems should adapt to and exploit their context of usage and they should be seamlessly integrated with other computer-based activities.

Furthermore, existing systems are of limited use since they are often tied to specific sets of underlying technologies, like networking technologies, or cannot easily adapt to changing requirements, like user preferences. Again, next-generation systems will need to provide means to accommodate different operational environments.

Reuse and Development Support

So far, little care has been taken to reuse components when designing and building new conferencing or groupware applications; nearly every system is built from scratch. Specific groupware development libraries or even development environments are still in an early stage and do not address all the problems (reuse, customization, integration, adaptation) sufficiently.

Sophisticated development environments will have to provide mechanisms to adequately model and design cooperative-software solutions in the overall context in which these solutions are supposed to be used. This requires appropriate syntactical and semantic support throughout the whole software lifecycle.

Media Usage

Today, the use of multimedia in conferencing systems is both transparent and transient.

Transparent means that media streams only supplement the shared workspace with communication channels. The rest of the system is unaware of their existence and their computer-supported coordinated use is at best limited to inter-stream synchronization (i.e., lip-synchronous presentation).

There is another interpretation of the word “transparency” that can in fact help to ease the construction of multimedia systems: Until now, hardly any attempts have been made to abstract media usage from its concrete representation. Future development systems should rather concentrate on real-world semantics like “conversations”; the decision about the actual communication media types could be deferred until runtime. This would also help to model systems that may use special devices or implement multi-modal interfaces.

Currently, audio and video information is also transient, since the data is simply lost after presentation and is not stored in a persistent way. There are three ways to improve the effectiveness and sophistication of media usage in collaborative computing.

First, the value of this conversational data can be improved by making it persistent. Then, the data could later be retrieved to trace back the steps that led to a decision. Today, it is still too costy to store all transient data and later retrieving the relevant parts of it.

These costs can be reduced if the system is really integrated into its context of usage. Then, contextual information can be used to structure and index the media data (e.g., associate data with the subject of the meeting, roles of participants, etc.). This context information can also be used to restrict the media recording and storage to certain periods within a session.

Storage requirements and retrieval costs can be further reduced if systems will be able to extrapolate, i.e., to extract and store higher-level descriptions from the raw data streams.

All in all, sophisticated and integrated use of multimedia data can lead to systems with radically new interfaces and interaction techniques that are better suited to their environment.

6.5 Research Agenda

This session has outlined problems and research topics in the areas of heterogeneity and interoperability, underlying system support, and development support. These topics are summarized as follows:

Collaboration Systems Architecture

To combat heterogeneity a flexible modular architecture has to be developed that provides the necessary abstractions to support the great variety of conferencing models and scenarios. Issues to be addressed here are:

- What are appropriate control models for computer-based conferencing?
- What should the protocols look like to support the range of different collaboration policies?
- What are efficient mechanisms for intra- and inter-site state distribution (application-level multicast and distributed messaging)?
- What are appropriate description methods to characterize system capabilities and requirements?
- What are proper communication standards?
- How do the system components scale with a growing number of participants?

Quality of Service Models

Conferencing systems pose special requirements on the network that need to be shaped in appropriate quality of service models. Issues to be addressed here are:

- How can applications adapt to variations in the service quality available from the network?
- What are appropriate signalling techniques to re-negotiate QoS during session lifetime?
- How many classes of services are needed?
- How many classes of services can be managed by the network?

Operating System Support

Today operating systems are not very well suited for multimedia real-time conferencing. Issues to be addressed here are:

- What are suitable scheduling policies for real-time multimedia?
- What is a good model for multimedia system services?

Collaboration Metaphors

New metaphors are needed that properly reflect the nature of computer-supported collaboration. Issues to be addressed here are:

- How should computer-based collaboration be integrated into the desktop?
- What GUI enhancements are needed to accommodate collaboration aware ness?
- What floor control policies are needed to better reflect group activities?
- Do we need new interaction techniques for computer-based conferencing?

- Are there any computer-based conferencing metaphors/patterns that go beyond physical conferencing situations?

Communication Quality

User acceptance of desktop conferencing systems heavily depends on how well they reproduce face-to-face communication. Issues to be addressed here are:

- What hardware/software is needed to support hands-free communication?
- What hardware/software is needed to support eye contact?
- What are efficient ways to minimize communication delays?

Development Support

Future development of collaboration systems should be based on software technology that can deliver customized solutions. Issues to be addressed here are:

- How can collaboration systems be customized and integrated with their context of usage?
- How can collaboration systems adapt to the operational and user context?
- What is needed to support reuse and development throughout the whole software lifecycle?
- How can media types be used in a persistent, integrated fashion?
- How can media transparency be achieved?

4 Multimedia Interfaces

Stefan Noll, Fraunhofer-Gesellschaft, Germany

4.1 Survey (White Paper Presentation by Steven K. Feiner)

What are the trends in multimedia user interface research? This session provided an overview of current and future research in two key areas: virtual environments and ubiquitous computing.

Among the most active topics at recent conferences on Human- Computer Interaction are computer supported cooperative work, multimedia, and intelligent interfaces. Most of this work takes the traditional desktop computing environment as a given. It is important to examine a companion set of research areas that go beyond existing hardware technology to ask how people will interact with future computers that may be quite different physically from those we now use. Two major paradigms that will strongly influence how we interact with computers are virtual environments and ubiquitous computing. Virtual environments are synthesized worlds created by coupling three-dimensional (3D) interaction devices and displays with powerful multimedia computers. Ubiquitous computing describes a future in which we are surrounded in our everyday life by a multitude of computers which unobtrusively aid us in performing our tasks and improve our quality of life.

4.2 Position Statements

4.2.1 James Foley, Darin Krasle

The major concern of developers should be to make multimedia systems usable. The end-user perspective is more important than the technology issues. One of the hard-learned lessons of decades of software engineering is that a lack of attention to the actual users and their needs can lead to failure. This danger is menacingly present in the developing field of Multimedia Systems since the technology is advancing so quickly that new product offerings are based on features and functionality, a situation known as "technology push", rather than on the actual needs of the users, known as "user pull."

Plenty of functionality is available as systems become increasingly more powerful, but the major problem is not what these systems can do, but rather how they can be used effectively. More extensive research is needed on metaphors, interaction techniques, device technologies, ergonomics, models, and authoring tools. The existing body of knowledge in these areas is incomplete and inadequate with respect to emerging technologies.

Metaphors determine how we think about the systems we use. The 2D desktop metaphor cannot be extended to a 3D environment by simply adding a third D to a 2D interface. It is as yet unclear as to which metaphors can be extended to work in 3D virtual environments and where new metaphors must be devised.

Interacting in such environments creates new classes of problems requiring new interaction techniques to be developed. Traditional techniques of using devices to input meaningful semantic units of information will not work in "opaque" environments such as Virtual Reality. The Windows-Icons-Menus-Pointing (WIMP) interfaces common today will prove inadequate. The "missing media" of speech input will play an increasingly important role in interacting with computers, especially in Virtual Reality and Ubiquitous Computing environments. Humans find speech to be an effective and often preferable communication medium, but it is difficult to uncouple from other more subtle channels such as pointing and gesturing. Giving directions over the phone is not quite as easy as helping someone with a map. Pointing is a useful thing to do while speaking. Sound has shown to be a powerful cue for computer-supported conferencing, games, visually impaired users, and status feedback in systems. The effectiveness of communication can be enhanced through use of multiple channels of interactions and the ability to translate or "trans-code" between them. It is important to identify such useful and meaningful combinations of interactions. After all, in order to empower computers to understand the complexities and interplay of human communication channels, humans must first themselves have an understanding.

Ergonomic issues must be addressed when new environments are considered. Humans are used to having consistent sensory input. Virtual environments attempt to deceive some of the senses to create an illusion, but what degree of fidelity is required to make this convincing? Will too little or too much realism cause people problems, or is there a range of acceptability? What are the most effective cues to support the feeling of immersion? How immersive must immersive interfaces be and how immersive can they be?

Model based designs are becoming popular for keeping track of useful explicit knowledge. Application models represent information that is useful in the design of a system as well as in its operation. At the design end of the spectrum these models support features such as design guidance, scalability, automatic generation of navigational views and user interface components, and control execution sequencing. At the run-time end of the spectrum models may be used for custom help, media "trans-coding", context identification for speech recognition, and quality of service demand management. In fact, application models allow the distinction between design and run-time to become less clear due to their declarative nature. User models have become important in providing users with interfaces that suit their specific needs. Knowledge about the users and their characteristics are modeled and can be used to customize the interface presentation and determine quality of service needs and preferences. Device and environment models contain information concerning the computing environment. Such knowledge is useful in determining the capabilities of the system to provide scalability, media "trans- coding", adaptation to the user, and quality of service.

One of the major impediments to multimedia systems is creating content. Authoring tools are one of the critical enabling technologies. Poor tool designs and the small number of product offerings stem from the fact that the main users of such tools are content specialists, not the technologists that design the tools. The key issues are the cost in terms of time to author and the quality of the results. Learning from other disciplines such as Rhetoric, Psychology, Educational Technology, Graphics Arts, and established media such as Movie, Television, and Drama will aid in the development of good tools. After all, the people in these fields have tremendous experience in creating content with only modest tools.

The design of multimedia systems and multimedia content also needs to draw on the budding field of Multimedia Rhetoric, which has been around since the early days of printing, in the combining of text and illustrations, but is only recently gaining recognition. Multimedia Rhetoric determines the manner in which different media may be used effectively and takes the goals and needs of both the creators and users, as well as the situation itself, into account. This has ramifications on many aspects of presentation including visualization, indexing, navigation, and metaphors.

The current proliferation of the World Wide Web shows much promise for the utility of multimedia systems. It is an example of a mixed-blessing allowing us to explore the benefits and pitfalls and brings up a host of research issues through its weakness. One might envision it as the FORTRAN of hypermedia systems which will ultimately lead to better paradigms.

4.2.2 Edward Fox

Multimedia systems should have usable interfaces that allow their users to efficiently and easily carry out tasks. Those interfaces should be scalable, allow media integration, and be dynamic. Developing such interfaces in the general case is a large, varied, and difficult undertaking. This needs research with older types of interfaces too. This paper attempts to reduce the problem to manageable size by drawing examples and focusing on an important class of multimedia systems and the corresponding set of matching tasks: those relating to digital libraries.

Digital library (DL) is now a grand challenge application in USA. There is great interest in France, Sinagapore and Japan also. DLs will contain multimedia forms of all types and in large quantities.

4.2.3 Research Items

The big issues are:

- authoring systems
- quality of service (QoS)
- information retrieval
- adaption

Research agenda for speech input:

- feed in context of appl. to improve recognition rates
- integrate into UI SW toolkits as a firstclass concept
- understand when/how to use it in place of current techniques, but at same time having appl. not require speech input (use in meeting)
- understand how to design appl. for speech

Research agenda for sound output:

- integrate into UI SW toolkits as a firstclass concept
- understand what types of sounds to use

Authoring tools: Quality of content

- higher-level constructs informed by educational technology and multimedia rhetoric
- rich model of data semantics
- hints for how to accommodate different QoS and devices
- multiple level support of hardware and networking capabilities
- uniformity across documents
- automatic generation

Authoring tools: Presentations as evolving data collections

- validation
- dangling references
- revision control

- data driven authoring tools

WWW/Mosaic

- semantic descriptors
- creating navigational views
- tools to create and maintain html structures

Part II:
Position Papers

A Multimedia Systems

A1 Evolution of Multimedia Systems on the Internet

Stephen Casner, University of Southern California, ISI, USA

The Internet is a key technology for multimedia, both influencing the purpose and structure of multimedia systems and changing to meet their requirements. Experience with real-time media on the Internet highlights the need for dynamic and scalable multicast distribution, which the Internet does well, but also the need for privileged service for real-time traffic. Work is underway to develop traffic control and reservation mechanisms for real-time service to meet the needs of multimedia traffic. Appropriate charging models must also be developed.

A1.1 Introduction

The session topics in this seminar cover the various technologies needed to "make multimedia happen." I believe we can rely on the vendors to build the necessary hardware capabilities into their platforms, though they can still use some help with coding algorithms that are more packet oriented and with the system support required to implement real-time applications (good timer support, low latencies). However, vendors aren't as likely to address the scaling or interoperability issues. In particular, I believe the Internet is one of the key technologies required for interoperation of multimedia systems on a large scale.

A1.1.1 Internet Influence on the Purpose and Structure of Multimedia Systems

I have a deep fear that the purpose of multimedia systems, as viewed by the investment sources, is no more than interactive home shopping and endless repeats of movies. I see the flexible communication enabled by the Internet as a means to achieve a better outcome. The usage-insensitive charging model widely employed in the Internet may encourage a wider variety of sources if it is carried over to multimedia.

The Internet communication model also contributes an idea that fits well with what I believe is the appropriate structure of multimedia systems. That's the notion of smart end systems, versus the dumb terminals of telephony. For high-performance multimedia, distributed processing will allow us to take advantage of the power of silicon to put processing in front of the users, as has been the case in general-purpose computing. For set-top boxes, the transition will take a bit longer to allow users' expectations to rise and for costs to fall.

A1.1.2 Requirements for Internet Support of Multimedia

Non-real-time multimedia data can be handled just like any other data. One impact of image data in the World Wide Web is that the bandwidth is much higher than it otherwise would have been, raising the ante for joining the Internet. Images also generate broader appeal, leading to more use and still higher bandwidth, but otherwise the protocol requirements are no different.

Support for real-time multimedia data requires more significant changes in the underlying basic design of the Internet. For example, resource reservations require increased state maintenance in the network routers. Therefore, the real-time issues will be the focus of this position paper.

A1.2 History of Real-time Media on the Internet

Twenty years ago, before the Internet existed, the feasibility of packet voice was demonstrated in experiments on ARPAnet. Continuous voice signals were digitized, compressed, chopped into pieces, sent over the network with varying delay, and reconstructed at the receiver. While improvements are still being made to this process, I think it is safe to say we basically know how to do it.

Around 1980, with the Stream Protocol (ST), applications were able to specify to the network what resources were required to insure minimum delay and loss. That mode of operation struck a balance between the fixed bandwidth allocation of a circuit and the inherent multiplexing flexibility provided by packets. ST also provided the capability to deliver a packet to multiple destinations. It should be noted that ST and is successor ST-II have now gone into production use, though only in certain networks and with the number of participants in a session typically limited to ones or tens.

Then, over the past two years, the use of real-time audio and video over the Internet grew to a much larger scale through the deployment of multicast Internet Protocol (IP). Steve Deering developed the multicast extensions to IP in the late 1980's, but deployment languished due to insufficient demand.[1] The necessary jump-start was provided by a crazy experiment to transmit live audio and video using IP multicast from meetings of the Internet Engineering Task Force (IETF). This caught the attention of the Internet community and led to the creation of the Multicast Backbone (MBONE), an experimental deployment of multicast routing software in workstation-class machines to form a virtual network layered on top of portions of the physical Internet. The MBONE has grown to over 1000 networks and subnets in 20 countries.

IP multicast routing is now being implemented in production IP routers. Over the next few years, it is expected that IP multicast will become a standard network service and the MBONE will fade away as a separate entity. This is a key step because the full network bandwidth will then be available to multicast traffic.

There are several factors that make the Internet an attractive choice for multimedia communication and that have fueled the growth of the MBONE:

- It is existing infrastructure with worldwide span.
- Many workstations are already connected with high-speed interfaces.
- It provides scalable multipoint distribution without hubs.
- There is currently no explicit usage-sensitive charging in most of the Internet.

A1.2.1 Events on the MBONE

During its two-year existence, the MBONE has carried a wide variety of events. The IETF meetings have been among the most popular, with a remote audience that has grown to about 600 participants in 16 countries, which is similar to the number of local attendees. But there have also been dozens of other workshops and symposia, both in the networking area (RIPE, JENC, INET) as well as other areas of science completely separate from networking. One example was the Pancreatic Islet Symposium which included was a presentation made remotely from Massachusetts to the assembled audience in California. Other popular events have been the broadcasts of live coverage of several missions of the Space Shuttle, including the Hubble telescope repair, and the annular solar eclipse as observed at from Purdue University.

The MBONE is not just for big events, though, as it is now also used routinely by working groups and for small, private meetings. Furthermore, it's not just for tech-

1. Paul Mockapetris, now chairman of the IETF, commented to me a few years ago that IP multicast was a good idea but it seemed like it would never get off the runway.

nical communication. There is a permanent music channel called “Radio Free Vat” and special events such as a multimedia performance called “Timewave Zero” from Sweden, a lecture on fine art holography from California, and a broadcast of the film “WAX or the discovery of television among the bees.” Two unusual events were a retirement party for Elizabeth Barraclough, Computing Service director at Newcastle University, and the installation by crane of a new Cray computer at DKRZ in Germany. In addition to audio and video transmissions, the MBONE also carried oceanographic telemetry data for scientific visualization in the JASON Project, and daily carries weather satellite images and multicast distribution of Netnews.

A1.2.2 Importance of Bi-directional Communication

It should also be noted that the IETF meetings were not simply one-way broadcasts. Not only were 600 participants able to listen and watch, but they were also able to speak back to the assembled group if they wished. In these sessions, the back channel is always ready, so the remote participant may begin speaking at any time without the overhead of setting up a call. The biggest problem with remote participation is convincing remote participants to speak. This problem has been observed in the current instructional television system at USC in which remote participants must place a phone call to ask a question. It’s critical to make participation easy.

Maintaining the latent back channel is feasible with packet switching because no voice packets are sent when the remote participants are not speaking, so the resources consumed by the latent back channel are quite low. This is especially true with connectionless IP multicast communication. There are some resources consumed, in particular if bandwidth is reserved, but some researchers believe bandwidth reservation will not always be required in a network with sufficient provisioning and with adaptive receivers. In particular, it may be possible to begin transmission with best-effort service during the reservation setup period.

A1.3 Problems to be Solved

It seems clear that the Internet could easily be overloaded with real-time audio and video traffic if there is no mechanism to control its use. Furthermore, the best-effort delivery service that has always been fine for data is only sufficient for real-time media when there is no congestion.

To solve these problems, work is underway by researchers and in IETF working groups to develop traffic control mechanisms for Internet routers to provide privileged service to real-time traffic. To invoke those traffic control mechanisms, other working groups are developing protocols that allow applications to reserve resources.

When there is insufficient capacity, reservation requests will be blocked, thereby avoiding overload at least within the privileged service classes. For the service to be satisfactory to users, such blocking should be rare, which means the overall capacity of the Internet must be increased. However, the cost of the additional capacity must be recovered, and there should be some form of feedback to the end user so that capacity is not wasted. For example, there might be a higher price for real-time service versus best-effort service. Since many people consider the Internet's usage-insensitive charging to be one of its strong points, it seems clear that the service and charging models are important areas for future work.

A1.4 Architectures for Scalable Real-time Media Distribution

If the experience with the MBONE is any guide, one can see that there is a wide range of potential sources of material with a pent-up demand for some appropriate transmission mechanism. It is hard to say how much of that demand would dry up if the service were not "free." On the other hand, if there are receivers interested in listening to those sources, then it should be possible to recover the cost of the service. For example, in the future it might become common for the Distinguished Lecturer Series at a university to be transmitted to a national or international audience. Going a step further, we may find the regional remote classroom systems used by a number of universities today expanding to become global.

Many of these applications require wide-area distribution on a relatively large scale. Today, that kind of service is mostly provided by broadcast over a shared medium, either radio frequency transmission or cable channels. To expand beyond the limited channel capacity of these systems requires going to switched communication. The high bandwidth available with fiber optics should be sufficient to satisfy any demand so long as the signal from each source is multicasted to the appropriate destinations and not broadcasted to all possible destinations.

To allow for highly dynamic distribution, a labeled multiplexing system is more appropriate than a circuit-switched system. This contention is supported by the development of ATM. However, there may be considerable debate about whether a virtual circuit model or the "soft state" approach of IP multicast is more appropriate. I believe either could be made to work, but that as the session size scales to 10^4, 10^5 or 10^6, it is increasingly important to distribute the workload of session management. Building sufficient capacity into the source node to track all the receivers may not be practical and is not necessary. Various IP multicast routing schemes are in use or in development, but they share the characteristic that receivers can join by hooking in to a nearby point on the distribution tree without requiring any interaction with the source node. Sources may still need to be in control of charging, for example restricting access via encryption, but that work can also be distributed so that receivers need not interact directly with the source node.

A1.4.1 Internet, Telephony and ATM

Internet has really caught fire over the past year or so. A larger user base will provide support and motivation for growth of the Internet as a whole, while a shift from government to commercial support with allow more flexibility to support new markets and new applications such as the real-time media. The larger user community will also drive the scaling requirements mentioned earlier.

Still, it seems unlikely that Internet will replace the telephone network any time soon. However, it's not an absurd idea: if there is a high bandwidth connection into a business or residence, and telephone service can be provided over that link for a very small incremental cost, then it won't make sense to pay for separate analog phone service.

The high bandwidth connection might be ATM, but not necessarily so. Although ATM will be a major networking technology, it seems unwise to assume homogeneity throughout the network. The Internet will link together existing and future network technologies as they are developed.

ATM is not necessary to provide real-time service. Dave Clark has pointed out that it does not matter whether packets or cells are being switched; ATM and Internet switches need the same traffic control mechanisms which have not been fully developed in either system [1]. ATM may also prove insufficient if its multicast capabilities don't meet the flexibility and scaling requirements of applications such as those described here.

A1.5 Conclusion

I believe the Internet is one of the key technologies required to make multimedia happen. The wide variety of events seen on the MBONE portion of the Internet over the past two years demonstrates a demand for the dynamic and interactive multicast distribution that the Internet and IP multicast provides. However, there is significant additional future work required to provide real-time service in the Internet. It is also clear that one of the biggest problems is only partially technical, and that is to develop the appropriate charging models to go with the new services.

Acknowledgements

This research was sponsored by the Advanced Research Projects Agency (ARPA) under Ft. Huachuca contract number DABT63-91-C-0001. The views and conclusions in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of ARPA or the U.S. government.

References

[1] Clark, D., Presentation at Internet Engineering Task Force Meeting, Columbus, Ohio, March 1993.

A2 Multimedia: Markets and Evolution

Jonathan Rosenberg, Bellcore, USA

This paper argues that the multimedia market will experience tremendous growth once networked multimedia is widely available. This will require significant investment in new infrastructure. This implies that the initial (and, thus, key) markets will be those where significant network infrastructure is already available. Based on this reasoning, the paper discusses the initial applications, the secondary markets and the key factors that will influence the evolution of multimedia.

A2.1 Key Markets for Multimedia Technology

Although the multimedia market is expanding rapidly, explosive growth awaits the widespread availability of *networked multimedia*. Networked multimedia is the delivery of multimedia information over networks, be they LANS, MANs or WANS, public or private.

Significant penetration of networked multimedia capabilities requires careful coordination among multiple complex components, including network hardware and software, computing devices, applications, and systems and operations software. In addition, content providers must have the capability to cost-effectively produce large quantities of compelling digital material.

The widespread availability of networked multimedia is expected to significantly affect many aspects of our lives in such areas as education, business, health and entertainment. People envision students attending classes from remote locations, just-in-time training delivered to employees' desktops, doctors using video to perform hospital consultations from their homes and people shopping electronically from the comfort of their couches.

It is easy to imagine the wonderful things we will be able to do once networked multimedia is widely available. Unfortunately, networked multimedia is limited today, typically found only in research laboratories and some businesses. It is by no means widespread (being virtually nonexistent, for example, in residences).

A2.1.1 Limiting Factors

Before discussing what the key markets will be, it is essential to know the primary factors limiting the widespread availability of networked multimedia. The key limiting factor is the *infrastructure problem:* the fact that a lot of "pieces" need to be in place before networked multimedia becomes feasible.

This can be seen by analogy to the telephone network. Consider what the value of a telephone would be without an available large infrastructure, including access networks, inter-office networks, switches, operations and maintenance systems, operators, etc.[1] Similarly, an end-user networked multimedia device is only useful if there is a correspondingly large infrastructure in place. This infrastructure must include

- specialized access and inter-office networks,
- specialized operations and maintenance systems,
- multimedia servers,
- directories.

Note that although access and inter-office networks, as well as operations and maintenance systems, exist today, the new systems will be significantly different. For the most part, production versions of these system do not exist and it will be a significant challenge to develop them.

In addition, It's not clear that a simple evolution of computers will support "true" multimedia in the near future. A radical change may be required. Unfortunately, most proposed solutions have suggested support for digital video by bypassing the bottlenecks (e.g., I/O channels directly from the network interface card to the video decompression card, or an interface from the video decompression board to the frame buffer, in both cases bypassing the system bus). But these solutions sacrifice generality, since they necessarily limit the operations that can be performed on media streams. We would like to find solutions that will constrain developers as little as possible. The bottom line is that the type of information that computers are being asked to process is fundamentally different, nd this may require fundamental changes to computers.

---

1. The terms in this list are from the world of telephony, though most of them have analogues in the more familiar data networking world. *Access networks* are the facilities used to connect users to the larger capacity, backbone network (known as inter-office networks in telephony). For residential customers, access networks are also known as "local loops". In data networking, access networks are often called "drops". *Switches* are large machines used to dynamically and rapidly transmit information among end points, and are analogous to routers and bridges in data networking. *Operations and maintenance systems* are software systems used to control the operation of telephone networks. This role is typically distributed throughout data networks.

A2.1.2 Initial Applications

Networked multimedia applications will first achieve significant penetration in those markets in which some appropriate infrastructure is already present. These will be the key markets, since technology investments in these markets will serve as an investment to lower the cost of networked multimedia. Lower costs will support the building of the new infrastructure to support networked multimedia applications in new markets.

Currently, there are two venues in which networked infrastructure is available: businesses (many of which have significant LAN and WAN environments) and the Internet community (which consists of WAN backbones connecting LANS and computers, and overlaps with many business networks). For these markets, the initial applications will be

- business: multimedia email, business presentations, information access and distribution, and computer-supported collaboration,
- Internet: multimedia email, information access and entertainment.

A2.2 Evolution

The two largest non-business markets are the educational and residential markets. In both of these areas, significant construction of infrastructure will be required.

Once penetration has begun in the business and Internet markets, networked multimedia will move to these other markets. Initially, the infrastructure will be constructed where network access can be concentrated to serve many people from a single location.

This implies that education will likely be the next market, since access can be concentrated to schools, providing simultaneous access to multiple users. Applications for educational use will include large-screen videoconferencing, delivery of "canned" educational material and information access.

The residential market will be the last to be penetrated, since it requires the largest investment in new infrastructure, and since each home provides only a small increment in the number of people with access. The residential market requires not only new network infrastructure, but also significant support or administration and maintenance. These are functions that businesses and the Internet community provide for themselves.

A2.3 Key Influences

The key influences on the evolution of multimedia are no longer technical. (This does not mean that there are no interesting technical issue to be solved, just that other factors are dominant.) Networked multimedia is now believed to be (very) big business. The bulk of the work on networked multimedia has moved out of research labs and into development shops. This means that the dominant factors are now economic and not technical, since work follows money. With his in mind, the key questions to consider are

- Which players have the money to invest in infrastructure?
- Who will pay for networked multimedia applications and services?
- What applications do customers want and which applications will yield business advantages?
- What effect will the politicization of networked multimedia (in the form the National Information Infrastructure, NII) have?

The answer to who the big players will be is being played out every day, as seen in newspapers. Who will pay remains quite unclear and there is much jockeying for position and alliance building to try to sort out this issue. Although there is much speculation about the applications that people want, there remains considerable consternation, since we will only know for sure once systems are deployed. The final issue, the role of the government in all of this, remains a wildcard.

A2.4 Influencing the Evolution

Computing and telecommunications technologies are advancing at a blinding pace. Processing power continues to become faster and cheaper. Network bandwidth appears virtually unlimited in the long run and the appearance of the NII seems inevitable. On the other hand, new network infrastructure (and its supporting pieces) is expensive to deploy and appears more slowly than any of us would hope. The faster that appropriate network infrastructure appears, the sooner that networked multimedia applications can be deployed. The more applications that are deployed, the sooner that manufacturers will develop cost-effective hardware, etc. Deploying appropriate network infrastructure is the fastest means for breaking the chicken-and-egg cycle that is preventing the widespread availability of networked multimedia applications.

The primary technical issue limiting the speed of network deployment is the lack of standards supporting interoperability. Such standards are particularly needed to support interoperability among end-user devices, content providers and multimedia servers. Without these kinds of interoperability, users will be faced with incompatibilities among hardware and software, which will greatly limit purchases. Consumers will typically not tolerate multiple, incompatible formats, as evidenced by the VHS-Beta VCR format wars and the disappointing sales of consumer CD-ROM based entertainment machines. In addition, content providers are begging for compatibility to avoid the current need to gamble, by picking a small number (often, one) of formats for which to produce titles.

A2.5 How Will Multimedia be Different Tomorrow

Today, the technical factors constraining networked multimedia are available network capacity and capabilities, and low-level technologies, such as synchronization, compression, etc. Over the next 5-10 years, we expect that great progress will be made in these areas, effectively solving many of the outstanding issues.

The result of this will be plenty of multimedia information available for access via networks. These networks will support vast numbers of people and provide adequate capabilities to deliver multimedia information with high quality.

At this point, the important technical issues will revolve around how people will find the information they want. This includes questions about how information will be indexed, catalogued, located and filtered. The hot areas for research in the next 5-10 years will be

- indexing of multimedia information,
- automatic classification of multimedia information.
- semantic searching of non-text media,
- techniques for filtering information based on user preferences,
- technologies for personalizing and customizing multimedia information.

A3 Is There *a* Purpose of Multimedia Systems?

Daniel C. Swinehart, Xerox PARC, USA

This paper attempts to take a look below the grand vision of multimedia systems at some of the forces that are preventing the field from developing as well as it might, and at some steps we might take to improve the situation.

A3.1 Objective

The stated topic of this session: to explore the purpose of multimedia systems, has reawakened concerns I have had for some time about the labels that are applied to broad segments of our industry. My concern is that by focusing too much attention on a particular concept of a full-spectrum system, we will overlook the individual value of the components making up that system; and that we will furthermore overlook opportunities to exploit these individual components in other useful ways. In the remainder of this discussion I will draw on a number of observations I've made over the past few years in order to explore these two related concerns as they apply to the collection of applications and techniques that are commonly known as multimedia systems.

A3.2 How Can Such an Appealing Buzzword Get Us into Trouble?

Let us explore a few examples of the dangers of assigning catchy names (such as multimedia) to related collections of useful technologies.

A3.2.1 OON—Object-Oriented Nomenclature?

First, an example from another field, to give the flavor of the issue being addressed. In recent years an entire subfield of Computer Science has emerged around the concepts of object-based programming. Although few would argue the virtues of these concepts, I frequently worry that the emergence of conferences and periodicals devoted to making object-oriented programming the "high order bit" has unnaturally partitioned the community of practitioners. Proceedings of object-oriented systems conferences deal with many of the same issues of operating system performance, data structuring, language design, user interface concerns, etc., that are also addressed by practitioners who identify themselves with these various fields. Conversely, said practitioners make frequent use of object-oriented methods. Since these workers seldom appear at each others' conferences very similar approaches are developed and published in venues with disjoint audiences, leading to replicated work and lost opportunities for collaboration. Are we doing the same thing by insisting on a definition, let alone a purpose, for multimedia systems?

A3.2.2 What Would Anyone Want a Telephone for?

That is a question that Alexander Bell and his colleagues had to answer, but it sounds foolish today. Similarly, broadcast and cable-based radio and television have well-established uses in entertainment, education, and business. Related uses such as video conferences have also gathered strong followings as their costs have dropped. To me, it seems unnecessary to worry much about whether there's a market for these technologies. Yet these are some of the fundamental new technologies, along with emerging digital networks and recently developed storage mechanisms, that are combined to produce multimedia systems.

But what about the market for a Collaborative System for Coordination of Design Activities of Suspension Bridge Architects? Those who insist that a multimedia system must combine specific features into a specific artifact for a sharply-defined purpose may find that they have a selling job on their hands. Nevertheless, the individual components of such a system, if not hamstrung by the user interface for the Architectural Application, remain inarguably useful, for suspension bridge architects and for others.

A3.2.3 We also Make a Consumer-based Multimedia Version. . .

I attended a recent demonstration of a powerful professional video/audio production system, applying the performance and convenience of all-digital techniques to the production of television features and commercials, At one point during the presentation, the speaker mentioned that the company had produced a "multimedia version," a stripped-down unit intended for desktop use. This is not necessarily a bad thing, but it revealed to me one perception of what constitutes a multimedia system — an affordable, limited-quality combination of features for the unsophisticated user. I would wager that this version is marketed in a way to inhibit ready upgrade to the full capabilities of the original. This company makes an impressive product whose “multimedia” version may be only an interesting toy.

A3.2.4 Look What We Had to Invent as Part of this Multimedia System!

Researchers who set out to build multimedia systems with new properties seem often to forget that many of the needed components might already be available. I have read myriad papers describing multimedia systems some or all of whose components were invented from scratch, or at least beginning at a lower layer than necessary. A typical prototype provides its own user interface, transmission and switching methods, session/floor control capabilities, storage formats, etc., even if its intent is to innovate in only one or two of those areas.

I speculate that developers work this way for several reasons: (1) the necessary components are not widely known; (2) the necessary components are not packaged for reuse (as implementations, protocols, or standards); or merely (3) the “Not Invented Here” syndrome causes them not to look very far for useful components. A variant (3) frequently arises in the large university or industrial research project, whose purpose is to produce numerous dissertations on a variety of topics related to some central theme; this can lead to the development of proprietary protocols or methods that others are unlikely ever to share.

A3.2.5 How Do We Know What to Build Unless We Know What it Will be Used for?

The providers of critical components for multimedia systems can also fall into the overdefinition trap. David Clark of MIT has captured this problem in an effective anecdote: In a recent meeting with telecommunications vendors, Clark says that he was asked some variant of the above question — trying to get to his telecommunications requirements by asking about specific applications. Quite reasonably, based on decades of telephone system experience, the vendors believe they must understand the communicating applications before they can design, configure, and provision their networks. Clark's response, and mine, is that two-plus decades of Internet experience have taught us that it is folly even to imagine the myriad uses for communication among geographically distributed systems. Like it or not, we can't usefully answer the question. What we can do in this instance is specify the characteristics that we expect our networks to provide, in terms of scope, addressing, bandwidth, delays, latency, and other qualities of service. Then we'll all have to be surprised together at what we really end up building. This has been referred to as the “Field of Dreams” approach to deploying new technologies.

A3.2.6 What Applications Motivate Your Creation of a Local ATM Environment?

At Xerox PARC, we're often asked why we are installing a high-performance local area network, based on ATM service directly to the desktop and to server hosts. The questioner no doubt expects a description of a flashy new multimedia application, such as the architectural design collaboration system spoofed earlier. I usually answer that I'd settle for the functionality of everyday cable television — but to *every* office, not just a few specially-provisioned ones; and by the way, transported over the same wire or coaxial cable or fiber that brings file access, electronic mail, remote computing, information web browsing, and all the other familiar internetwork services to my office.

I do not argue the value of the more ambitious systems, but I assert that value can also be added by supporting more mundane applications in an integrated fashion.

A3.2.7 Why Can't We Watch Our Multimedia Presentation Through Mosaic?

Well, it turns out that in some places, you can. But the point is that if we adopt too tight a definition for multimedia systems and insist on combining their functions into unfactorable monoliths, we decrease the opportunity for combining their capabilities in other ways.

A3.2.8 Concerns

Before going on, I'd like to summarize the concerns that motivate these examples. I am concerned that in defining a “purpose” for multimedia systems we will:

- produce a community of practitioners that reads its own papers and listens to its own talks, but fails to participate in the technical fields that support the individual components of their systems.
- fail to acknowledge the value of the more mundane applications, taken separately, that can be carried usefully on an integrated infrastructure.
- overlook the extent to which these components, individually or in groups, can be combined with other applications to produce unimagined future products.
- overlook the individual challenges of supporting each medium robustly, efficiently, cost-effectively, and with careful attention to modularity.
- continue to develop new component-level solutions to old problems as each new system is built, delaying the emergence of widespread standards.

A3.3 So What Does It All Mean to Us?

Am I arguing that there is no purpose to multimedia systems, that we have nothing to do and should not have convened this workshop? Of course not. But I would hope that, if any of these concerns are shared, we can use them as guidelines for what we would like to accomplish.

Looking at our agenda, it is clear to me that the organizers are well aware of the fundamental components that comprise multimedia systems, and that these subjects are well represented. I might suggest that we add something here or there—for example, an explicit focus on authentication, security, privacy, and in particular the effect of firewalls techniques on performance—although I expect these issues will emerge as part of the networking, documents, and conferencing sessions.

What might be useful is to remove the word “Multimedia”from the prefix of several sessions, then decide whether each topic has been improved or damaged.

My primary interest is for us to accomplish something important with this workshop—to produce an actual work product, as well as share state and understanding.

I believe that our field is in need of some taxonomy and organization. Most likely, it is also in need of significant “shakeout.”Every week I read an announcement of another consortium of two to thirty companies which will bring order out of chaos. Each consortium of course introduces another incompatible set of standards and specifications. Sometimes this is done intentionally, for competitive reasons. Sometimes it's because different subfields, with no knowledge of each other, act independently. Example: file format and transmission standards have emerged independently from the office systems, multimedia, computer music, and professional production communities. And none appears to have anything to do with standards emerging from communities such as the Internet Engineering Task Force. But each is going to take over the world.

In the past, some notable workshops have produced important architectural overviews that helped their respective fields mature. An exceptional example was the Workshop on User Interface Management Systems [1], held in Seeheim in 1983, which is generally seen as a watershed event by the user interface community. If we can come close to the impact of this event, then we really will have accomplished something. Someday people would refer to Dagstuhl as an event, as well as a place.

The agenda provides a good outline for this approach. I recommend we examine each topic with an eye towards addressing the following:

A3.3.1 Is There Much More Work to be Done?

We need to identify those areas where significant additional work is needed before reaching any sort of maturity — where it is much too early to encourage any sort of “shakeout” or to begin making constraining choices. In these areas, it is important to encourage the continued development, and to abide the multiplicities of approaches and the overlapping capabilities that result. Perhaps we can make an attempt to identify the challenges that must still be addressed.

A3.3.2 Is It Time to Choose?

But we should work equally hard to identify areas that we understand pretty well, those that are mature enough that we might reach consensus on selecting a small number of candidates to promote as standards, and then agree to help “market” these choices to the rest of the community. The user interface community seems to have reached a degree of stability that has produced benefits; this is the sort of thing I'm talking about. It is futile, and likely foolish, to try for a single candidate, but we may be able to reduce proliferation of additional approaches when their development adds nothing new to the mix. The goal here is to identify and promote well-established, leading contenders for widespread acceptance.

A3.3.3 At the Least, Let's Make a List!

The organizers have gathered a group of the most knowledgeable practitioners in the world. Together, we must have stored a reasonably complete taxonomy of the multimedia systems in existence, and of the components, protocols, and standards on which they are built. If we were to do nothing else but make this taxonomy manifest, and then publish the results in some widely available forum, we would be doing a great service to the causes expounded here. This roster could include information about which facilities are experimental, which commercial, which publicly announced but proprietarily held, and which openly available.

A3.3.4 Let's Also Think About Interfaces

Perhaps more important than what a component does is how it communicates with related components in a system. I don't have a coherent recommendation here, but if we were to take on any of these other challenges, we should give special attention to reducing the number and increasing the quality of component interfaces.

A3.4 Conclusion

Perhaps these points are old news to everyone, and I'm guilty of carrying Palladium to Salt Lake City or some such thing. Or maybe you believe my concerns are misguided. If so, thank you for bearing with me this far, and let's press on. This piece was contributed with the hopes that, if it were to strike a responsive chord with other participants, we might consider some of these concerns as we go about the job of considering the future directions of multimedia systems.

References

[1] User Interface Management Systems. Proceedings of the Workshop on User Interface Management Systems, G. Pfaff, ed., Seeheim, FRG, November1983. 224 pp. Springer-Verlag, Berlin.

B Media Encoding and Compression

B1 Video Decompression Should be Done in Software

Wolfgang Effelsberg, University of Mannheim, Germany

Most of the video compression schemes standardized today are computationally intensive (H.261, MPEG and the much abused JPEG). They optimize compression ratio at the cost of speed. They were developed under the assumption that special hardware will be available for video decompression on the target machine, and will be inexpensive when built in large volumes. However, decompression in hardware creates a number of problems:

1. Dependency on bus and display

An adapter board is developed for a specific bus architecture. For example, we cannot use our Parallax board, purchased for a Sun workstation, on a DEC Alpha machine or on a PC. This limits flexibility.

What is even worse is the dependence on driver software; an upgrade from SunOS to SOLARIS requires a new driver for the board, and timely delivery by the board manufacturer is not always guaranteed.

The adapter board also depends on the hardware of the display (CRT) in use. A slightly different display hardware can require the replacement of a chip on the video board. Experienced colleagues always by three cards if they need two, to have one extra to be in the mail.

As long as decompression is done in hardware there is no solution to these problems.

2. Frozen compression algorithms

In recent years incredible progress was made in the field of compression techniques. Compression ratios became much better, and the presentation quality was improved considerably. There is no end to this process. "Freezing" today's state-of-the-art by building and buying special hardware is not reasonable.

New and very promising techniques are still under investigation:

- Fractal compression schemes for images and video seem to yield excellent compression ratios, combined with high presentation quality.
- The use of wavelet transforms (i.e. mathematical wave functions with decreasing amplitude) rather than the popular cosine transform (based on a periodic function) are also very promising.
- Audio compression already uses the non-linear characteristics of the human ear; a mathematical model of the cochlea can be built, and the sender can eliminate redundancy which will not be heard by the receiver anyway. The use of similar characteristics of the eye is not well under stood.

From a researcher's point of view, it is still too early to prevent rapid progress by mass-delivery of inflexible hardware.

3. High-performance networks, disk arrays and processors

It was demonstrated recently (i.e. by the Berkeley MPEG player and in the XMovie project) that decompression and rendering can be done in software on modern machines, in good quality. The next few years will bring ATM and even faster RISC CPUs to workstations and PCs. Software decompression, delivered to the customer as an integrated part of the window system, will enable all these machines to participate in the multimedia world. When we evaluate compression ratio vs. universal availability and flexibility, the latter will become more important.

If software decompression is the future, special attention should be given to the compression algorithms. For example, algorithms without floating-point operations might run much faster in software. Dithering is another important issue for videos as long as 8-bit graphics adapters still dominate the market. The next generation of compression algorithms should be reconsidered under this aspect.

B2 Media Encoding and Compression

Bernd Girod, University of Erlangen-Nuremberg, Germany

The convergence of computer technology, telecommunications, and consumer electronics, nicknamed "multimedia", has lead to a variety of scenarios, where synchronous data streams such as audio and video are combined with asynchronous data such as text, graphics, or still pictures. Computers provide the flexibility to combine all these data interactively in innovative ways. Multimedia application scenarios range from desktop videoconferencing and computer-supported cooperative work to interactive entertainment networks, where movies-on-demand, video games, and teleshopping are provided.

Without question, the integration of motion video into multimedia systems is technologically the most demanding task, due to the high data rate required by video signals. For example, uncompressed video according to the digital television studio standard set by CCIR in their recommendation 601 requires an overall data rate of 216 Mbit/s. The data rate for an uncompressed HDTV signal is in the order of 1 Gbit/s. Most of today's workstation or PC screens already have a spatial resolution closer to HDTV rather than to standard television. Compare these rates to those handled in current computer systems. A magnetic hard disk will typically allow read and write access at rates up to 10 Mbit/s. An optical compact disk can be read at 1.5 Mbit/s. Ethernet typically allows peak transfer rates up to 10 Mbit/s, but of course the sustained rate depends on the network traffic and is much lower. ISDN is becoming widely available now, but its basic channel rate is only 64 kbit/s. A feature-length movie, stored in uncompressed CCIR 601 format, requires in the order of 150 to 200 GByte. Imagine a video-on-demand server with 1000 movies! Hence, digital video signals have to be compressed substantially to be stored and transmitted for multimedia applications.

The requirement of highly efficient data compression conflicts with several other requirements, some of which are general, others are specific to multimedia systems. Particularly, these include:

- Picture quality. Good spatial resolution, sufficient motion rendition, and the absence of compression artifacts are required. For many applications, a picture quality like the one provided by a VHS tape is sufficient, but a better picture quality would of course be preferred.

- Low delay. Videoconferencing applications require a low delay, otherwise the interaction between participants is seriously disturbed. For interactive video applications, based on stored video, short latency is desirable as well, but often less important.
- Scalability. Scalability allows for resizing of a picture in a window system and for graceful degradation in the case of network overload or fluctuations in available compute power.
- Access features. Fast forward and reverse (with visible picture), slow motion, and freeze frame features should be supported. Random access to individual frames is highly desirable.
- Editability. Ideally, we would like to cut and assemble video with the computer on the compressed bit-stream level, but decoding and re-coding of data right before and after a scene cut might be acceptable as well.

Video compression is a mature field today. Around 1980, codecs combining motion compensation and the Discrete Cosine Transform (DCT) first appeared. This basic ‘‘hybrid coding system" was analyzed, refined and optimized by many different contributors. Today we know, that other signal decompositions, such as the Discrete Wavelet Transform, are somewhat superior to the DCT, and vector quantization can outperform scalar quantization, as is typically used with the DCT.

Still, today's video compression standards are based on a combination of motion compensation and DCT coding. The rate required to encode a full motion video signal at VHS quality has come down from around 20 Mbit/s around 1980 to well below 1 Mbit/s today. For head-and-shoulder views typical for videoconferencing, rates can be substantially lower still. As algorithms are maturing, it has become harder and harder to lower the data rate even further. For the existing schemes, there is probably little room for improvement. New approaches, such as "fractal" compression based on iterated functions systems or ‘‘model-based coding" have yet to prove that they can lead to results superior to the classical waveform coding algorithms. It is doubtful whether bit-rate reduction for the VHS quality level can be advanced by another order of magnitude in the near future, but there is no fundamental reason why vastly improved image models could not yield another quantum leap in the future.

Motion compensation and DCT coding are the basis for the ISO MPEG-1 and MPEG-2 video compression standards. The MPEG-1 development was finished in 1991, MPEG-2 is basically done today. While it is not limited to the storage of motion video on compact disk, this application played an important role in the development of the MPEG-1 standard. Bit-rate and image size can be set flexibly, but typical parameters are 1.5 Mbit/s (including multiple compressed audio channels) and an image size of 288 lines x 352 pels. Frame rate is between 24 and 30 Hz. Unlike current television standards, MPEG-1 does not include line-interlace, so that display on non-interlaced computer screens is easily possible. The MPEG-2 standard is targeted at interlaced material, higher bit-rates, and new applications such as digital broadcasting. Typical picture quality corresponds to NTSC, PAL, or SECAM quality at 3 - 5 Mbit/s, and consumer quality HDTV has been demonstrated at 20 Mbit/s. Rather than a standard in the traditional sense, MPEG really provides a generic tool-box of techniques. Devices can successfully communicate by supporting a common "profile" of MPEG.

The similarity of successive frames in a video sequence is exploited by the MPEG algorithm, utilizing motion-compensated prediction. Rather than encoding each frame by itself, changes from frame to frame are encoded. If the luminance of the frame to be encoded can be predicted precisely from previously transmitted frames, the required bit-rate is low. In order to reduce the prediction error as much as possible, the frame-to-frame displacement is measured and used for prediction. The displacement vector field is also needed for decoding and therefore transmitted as ‘‘side information."

Motion-compensated prediction can be incorporated into different data structures to represent sequences of images. The simplest strategy is used, for example, in the ITU-T Recommendation H.261 for encoding of videotelephony signals at multiples of 64 kbit/s. This standard preceeded MPEG. With the H.261 scheme, the first frame is encoded in intraframe mode, i.e., without reference to other frames in the sequence. Then all successive frames are encoded in their natural order, using the preceeding frame for prediction. In principle, this strategy can lead to a codec with minimum delay. Unfortunately, it does not support random access or scalability. The problem is alleviated somewhat by the requirement that every part of the picture has to be transmitted in intraframe mode at least every 132 frames. In fact, successful transmission of packetized H.261 video over the LAN has been demonstrated, although robustness is problem.

MPEG overcomes the access difficulties built into H.261 by using the hierarchical frame structure. The encoder can declare frames to be one of three types. I-pictures are encoded in intraframe mode, i.e., by themselves. P-pictures are predicted from the previous I-picture or P-picture. B-pictures are predicted from a closest past I-picture or P-picture and the closest future I-picture or P-picture. Each potential entry point in the sequence should be encoded as an I-picture. Typically, there would always be a few B-pictures between P-pictures or I-pictures. For fast forward operation, one would skip some or all of the B-pictures. Fast reverse operation can either only use the I-pictures, or move backwards through sequences of I- and P-pictures, but play each of these sequences in fast forward mode. When encoding with MPEG, one has to find the right trade-off between random access capability and efficient data compression. With many I-pictures, coding efficiency drops since I-pictures do not exploit their similarity to other pictures. Having many successive B-pictures introduces a large coding delay. For decoding, this is not an issue, since the frames are transmitted or stored in the order that they are needed for decoding, rather than in chronological order.

Chip sets are available today that support MPEG or H.261 compression and decompression. In the future, these chip sets will have to compete with general purpose processors. Soon, the CPU power of a personal computer will be sufficient to run video codecs in real-time. Real-time software encoding and decoding on high-end PCs according to H.261 will become commercially available within the next 6 months. At the beginning of the next decade, a typical desktop multimedia computer is expected to execute more than 1 billion RISC instructions per second. Main memory of more than 1 GByte could be loaded into the machine, which would allow storage of more than an hour of compressed motion video in RAM.

When integrating motion video into an open computer system with a software-only codec, coding and decoding algorithms need extra flexibility to cope with the varying computation, transmission, and display resources of the system. This extra flexibility today is commonly referred to as ‘‘scalability." Scalability includes three issues:

1. Scalable image size

In a window system, the user wants to be able to resize the video frame flexibly. Of course, a large video window can display more detail than a small window. A naive solution would simply decode the image with all detail available, and then filter and subsample it to the desirable size. This strategy, however, is wasteful. Rather, we only want to read, transmit, and decode that portion of the information that is required to display the image at the desired size with satisfactory resolution.

2. Partial decodability

When transmitting video over multi-access packet networks such as Ethernet, the effective channel capacity is not known in advance. Rather, the video transmission is competing with other processes. As a consequence, the decoder has to be able to decode and display a picture from partial information. With network overload, the picture quality should degrade gracefully.

3. Computation-limited coding and decoding.

In a multitasking environment, there are several processes competing for the computational resources of the same workstation. Similar to the unknown transmission bandwidth on ATM networks, there is an unknown computational bandwidth that affects both coding and decoding. This issue becomes even more severe in multipoint videoconferencing situations, where multiple bitstreams have to be decoded simultaneously. Moreover, we might want to decode the same bit-stream with processors of different power. Even low-end workstations should be able to decode a given motion video bit-stream, even though picture quality might be reduced compared to high-end workstations.

The scalability requirements listed above are quite different from those of the classical telecommunication situation, where there are well-defined source and display formats, a fixed transmission bit-rate, and coders and decoders which are digital circuits designed and optimized for their specific tasks. The ITU-T H.261 video compression standard has no scalability at all. Even if only a few bits cannot be decoded, it might take several seconds until the picture recovers. MPEG-1 is somewhat scalable. If the channel or the decoder cannot keep up with the full bit-stream, we can leave out B-pictures without harm for other pictures. Unfortunately, B-pictures typically constitute the smaller part of the entire data.MPEG-1 has no mechanism for spatial scalability, i.e., pictures always have to be decoded to full resolution. MPEG-2 in principle includes several mechanisms for scalability, both spatial and temporal. Current research is developing layered scalable coding schemes suitable for various applications based on MPEG, but also based on other approaches. With software-only codecs dominating on the desk-top in the future, standards will probably play a less important role, since the incremental cost of supporting additional encoding schemes is very small.

Project Information

In the Image Communication Group of the Telecommunications Institute of University of Erlangen, we are addressing some of the open questions outlined above in ongoing research projects. A selection of relevant projects with responsible research staff member is listed below. Individuals can be reached by email via name@nt.e-technik.uni-erlangen.de

- Compatible improvements of standards based video codecs (F. Hartung)
- Fast algorithms for software-only video codecs (N. Faerber)
- Scalable video coding using hierarchical vector quantization (U. Horn)
- Model-based coding of moving video for very low bit-rate (K. Stuhlmueller)
- Computer-animated face-to-face communication (M. Link)
- Protocols for video and audio transmission in local area networks (K. Ben Younes)
- Mobile multimedia communication (N. Faerber, U. Horn, K. Ben Younes)
- Audio signal processing for videoconferencing (A. Stenger)

B3 Video Compression for Desktop Applications

Lawrence A. Rowe, University of California at Berkeley, USA

This paper discusses the current state of compression for digital video on the desktop. Today there are many choices for video compression that yield different performance in terms of compression factor, quality, bitrate, and cost. Users want a single low cost solution which, unfortunately, today is non-existent. Consequently, users will have to develop applications in an environment with multiple representations for digital video unless PC's can be assigned to dedicated applications. Alternatively, programmable compression/decompression boards can be used to solve the problem. Eventually, special-purpose hardware solutions will be replaced by general-purpose software running on desktop parallel processors which will be implemented by multiple CPU's per chip.

B3.1 Introduction

This paper presents my opinion of the current state of the art for compression for desktop digital video applications. Put simply, there are too many compression algorithms and standards and too few low-cost boards that implement the major standards.

B3.2 Current State of the Art

There are numerous video compression algorithms including: Apple's Roadpizza, Supermac's CINEPACK, Fractals, H.261, Intel's INDEO, motion JPEG (MJPEG), MPEG-1, MPEG-2, Sun's CELLB, and Wavelets. Users are confused by all these choices. They want to know which technology to use so they can make intelligent investment decisions. Unfortunately, the current situation is not very good because there is no single technology that can be used for all applications. For example, Apple's Roadpizza and Supermac's CINEPACK are designed for playback applications with software-only decoding, H.261 is designed for video teleconferencing, MPEG-1 is designed for low bitrate (e.g., 1.5Mbs) audio and video playback applications, and MPEG-2 is designed for high bitrate, high quality playback applications with full-sized images (e.g., CCIR 601 with studio quality at 4-10 Mbits/sec).

Users want one solution, but one solution does not exist. In the next couple of years, I see the following trends.

B3.2.1 MPEG on Every Desktop

Low cost MPEG-1 decoder chips will be on every desktop. Add-in boards cost around $350 today, and the next generation multimedia PC will have audio and video decoder chips on the motherboard. Manufacturers of video games and CD-ROM titles will use MPEG-1 video to add excitement to their products.MPEG hardware for workstations will be less readily available and more costly because these manufacturers can provide creditable software-only decoders for MPEG. Early experiments on software-only MPEG decoding showed that small-sized images (e.g., QCIF which is 160x120) can be decoded in real-time and medium-sized images (e.g., CIF which is 320x240) can be decoded in near real-time (16 fps compared to 24 fps) on RISC processors [Rowe, Patel, and Smith 1993]. Subsequent work by DEC showed that tuning the decoder to a specific processor can achieve real-time decoding of CIF images [Ho 1994]. Recently, HP released a software-only MPEG audio and video decoder for their HP Snake processors that runs in real-time on CIF images [Lee 1994]. The HP software uses special-purpose instructions added to the architecture that speedup Huffman decoding and 8-bit arithmetic operations (using saturation arithmetic). And, they use hardware to convert YCRCB to RGB and dither to an 8-bit color map. Color space conversion was done in software in the other cases which can be as much as 30% of the computation. Nevertheless, the HP software is impressive.

These experiments illustrate that software-only decoders will eventually replace all hardware decoders. I believe that it will be at least 4-6 years before hardware decoders for MPEG-1 are out-dated. By that time, hardware decoders for MPEG-2 which supports higher quality video and audio at higher bitrates will be widely available. Some users will upgrade to higher quality rather than continue with low quality at no cost. A general-purpose processor capable of MPEG-2 decoding on full-sized images (e.g., 640x480 or 768x576) will require multiple processors.The biggest problem with MPEG is the cost of encoders. High quality, real-time encoders cost between $50K and $500K. Almost all high end encoders use parallel processors, either general-purpose supercomputers (e.g., IBM) or custom-designed video processors (e.g., CCube). Lower quality real-time encoders for PC platforms that use fewer processors cost around $20K (e.g., FutureTel, Optibase, Optivision, etc.). While the cost of these low end systems will decline over the next couple of years, they will still be too expensive for most users.

B3.2.2 Motion JPEG for Editing

Non-linear video editors are typically used in broadcast TV, commercial post production, and high-end corporate media departments. Low bitrate MPEG-1 quality is unacceptable to these customers, and it is difficult to edit video sequences that use inter-frame compression. Consequently, non-linear editors (e.g., AVID, Matrox, FAST, etc.) will continue to use motion JPEG with low compression factors (e.g., 6:1 to 10:1).

Motion JPEG compression has also been used in some desktop video conferencing applications (e.g., Insoft) because affordable workstation boards that support real-time encoding and decoding have been available. Typical boards cost $4K to $10K. Motion JPEG boards are now being sold for PC's that cost $1K to $4K.

B3.2.3 H.261 for Video Conferencing

Video conferencing has been an active research and product area for many years. Although most commercial room-sized conferencing systems use proprietary standards, they are now adopting the H.261 ITU standard for video conferencing. Moreover, most desktop video conferencing systems are using H.261 (e.g., AT&T, Compression Labs, Intel, PictureTel, etc.). Most of these systems use ISDN lines, although a few are starting to support packet-switched networks. And, several research laboratories are developing software that uses H.261 boards on PC's and workstations.

B3.2.4 What's the User to Do

What is the user to do who wants to provide ubiquitous digital video, that is, video in all applications including email, documents, conferencing, hypermedia courseware, and databases? Users have two choices:

- Select one compression standard and try to acquire applications that will use it.
- Acknowledge that you need support for multiple compression standards.

My opinion is that users will have to make the second choice which means either a programmable compression/decompression board or multiple compression boards. Programmable boards exist, but they are not widely available, and they are expensive. In addition, vendors do not yet provide microcode for the variety of compression standards needed, but I believe that eventually the software will be readily available and relatively inexpensive. The question is will the software be available for programmable boards before parallel processors for desktops are available that can run general-purpose software.

In the meantime, users must develop applications that are open so that new compression technology can be introduced and so that real-time conversion is supported. For example, Quicktime from Apple and Video-for-Windows from Microsoft are the dominant storage systems for PC video. Both systems support multiple compression standards.

Better support is needed in applications to convert between different representations because most applications are closed. For example, a desktop video conferencing system should allow video transmitted in H.261 format to be converted to an MPEG stream so that PC users can view remote presentations.

B3.3 Research Problems

This section discusses some possible research problems.

Some researchers argue we need improved compression technology such as wavelet-based algorithms. Except in the case of wireless communication discussed below, I disagree. I believe that research should be directed to improving the existing technologies and developing improved implementations, systems infrastructure, and applications. Unless a new technology can provide significantly better performance (i.e., at least 2:1 improvement in space) than the current JPEG, MPEG, and H.261 standards, users will be better served by improving the existing techniques and applications.

Some proposed compression standards provide other services such as multiresolution sequences (i.e., different applications can request different sized images at different bitrates from the same compressed representation) and variable quality (i.e., different quality at different bitrates). While these features are reasonable to request, I do not believe you need a completely different compression technology to support them. The MPEG-2 standard has provisions, albeit somewhat controversial, for image size, quality (S/N ratio), and frame rate scalability. I believe it makes more sense to develop the technology supporting these standards than it does to propose a completely different technology unless you get the compression improvement mentioned above.

B3.3.1 Multiple Format Stored Representations

Suppose you wanted to develop a video server for a heterogenous computing environment that included desktop computers with different decompression capabilities (e.g., motion JPEG, H.261, and MPEG-1). The problem is what representation do you store. You could store one of these representations and then provide a real-time transcoder somewhere on the network that will convert between the different representations. Another alternative is to store a representation that makes it easy to generate any of these sequences. For example, there are differences in the block and macroblock structure of these streams, but it should be possible to devise a stored representation that can easily generate any of the representations. Here are a couple ideas:

- Store several motion vectors for a macroblock. For example, MPEG vectors can be arbitrary far away from the origin of the source block, they can be on half-pixel boundaries, and, in the case of B frames that can be forward, backward, or an average of a forward and backward block. H.261 motion vectors can only be +/- 15 pixels, they cannot be on half-pixel boundaries, and they can only be backward blocks. So, the idea is to store two motion vectors for blocks whose MPEG vector is not valid for H.261 and select the appropriate one when constructing the stream to be transmitted.
- Store the huffman encoded representations of frames and create the rest of the stream syntax on the fly. For example, an H.261 stream can skip up to 2 frames between every frame displayed and although there is a requirement to refresh every block within some number of frames, there is no requirement to include the equivalent of a complete frame (i.e., an MPEG I-frame). The H.261 stream could be easily generated from an appropriate MPEG-like frame structure similar to the one suggested above.
- Provide support for scalable H.261 and motion JPEG using the MPEG scalable representations. A shrewd data structure and efficient algorithm implementation (e.g., possibly using frequency domain operations [Smith 1994]) should produce a more flexible system.

B3.3.2 Perceptual Coding

Much work remains to be done understanding the human visual system and developing models that can be used to implement better coders. Surprisingly, perceptual coding of audio is ahead of perceptual coding of video [Jayant, Johnston, and Safranek 1993]. Today, most researchers are working on best possible coding with infinite time to encode. The target bitrates are typically 1.2 Mbs for CD-ROM and 2, 3 or 6 Mbs for video-on-demand. There are many other points in the design space. For example, suppose you wanted to encode CIF images on a typical PC and you were willing to produce a statistical guarantee on bitrate. The idea is to relax the bitrate requirement because real-time transport protocols are being designed to provide statistical guarantees, so why should the coder work hard to satisfy a strict bitrate bound when it may mean a significantly poorer picture. The coding strategy for this implementation will be very different than the strategy used in current coders. This idea is only of several ways to change the basic model.

B3.3.3 Multiple CPU/Chip Implementations

Future desktop computer architectures will use microprocessors that support multiple CPU's per chip. For example, a RISC processor requires 1M to 3M transistors. Chip technology will soon be able to put 100M transistors on a chip. So the question is how to use the transistors? One design will put many different processor architectures on a chip so that a system can run different software. Another design will put many copies of the same processor on the chip.

An interesting research problem is to understand the effect of different architectures on compression and decompression. One possibility, which is probably already being done in industrial research labs, is to look at high performance parallel decoders for HDTV images (e.g., 1920x1080) using general purpose processors.

B3.3.4 Continuous Media Infrastructure

There is currently no portable toolkit for developing distributed continuous media applications (i.e., digital audio and video) such as desktop conferencing systems, distance learning systems, distributed video playback systems. Many excellent research systems have been developed, but they are typically not distributed, and they support few hardware platforms and audio/video boards [Anderson, and Chan 1991, Gibbs, et.al. 1991, Hamakawa, et.al. 1992, Koegel, et. al. 1993, Rossum, et.al. 1993, Steinmetz, and Fritzsche 1991, Trehan, et.al. 1993, Hewlett-Packard, IBM, and Sunsoft 1993]. There are several standards groups and large companies trying to establish common architectures and protocols for developing distributed applications, but these efforts have yet to succeed.

The consequence is that anyone who wants to develop an application faces the problem of developing the infrastructure. Our research group has developed such an infrastructure, called the Berkeley Continuous Media Toolkit, that supports motion JPEG and MPEG video boards, several audio standards, and runs on a variety of platforms. It is based on the Tcl scripting language, the Tk interface toolkit, and the Tcl-DP package for distributed client/server computing. We have developed a network playback system [Rowe, and Smith 1992] and desktop video conferencing system using the toolkit [Chaffee 1994].

You might wonder how a research project at a university can compete with large companies. The answer is we cannot. However, by distributing our source code and working with other researchers we can build a common infrastructure. This approach has worked for CAD tools, Tcl/Tk, and the INGRES relational DBMS to name three examples from Berkeley.

However, we still need the equivalent of the PBMPLUS library for manipulating digital video data. The idea is to develop tools and libraries so that different researchers can experiment with components of the infrastructure and with applications built using it.

B3.4 Wireless Audio/Video Compression

Wireless computing links are very different than conventional communication links. First, bandwidth is limited (e.g., approximately 2 Mbs aggregate bandwidth in a cell). And, communication errors are inversely proportional to the power used on the portable device. Power is the scarce resource so algorithms and implementations that perform adequately with less power are better. Some researchers argue that portable devices should have limited computational power to reduce power requirements which means that audio and video compression must be very simple [Broderson 1993].

Compression algorithms that work well in this environment are an interesting challenge. Some people are looking at pyramid and subband coding using vector quantization. Vector quantization is simple to decode and pyramid and subband coding can be used to partition the stream into high priority data that will be sent with more power to reduce errors and low priority data that will be sent with less power.

Needless to say, this architecture will create many problems if the rest of the digital video infrastructure is dominated by the block transform coding standards as I believe it will be.

B3.5 Conclusions

Compression researchers have developed numerous technologies that have been used to develop a series of compression standards that will dominate desktop digital video. Today, and for at least the next 5-10 years, application developers and users face a difficult choice of which hardware and software to use. Eventually, desktop parallel processors will allow many different compression algorithms, implemented in general-purpose software, to be used.Many research problems remain but my opinion is that effort should be directed to improving existing implementations, software systems infrastructure, and applications.

B3.6 References

Anderson, D. P. and Chan P. 1991. Toolkit Support for Multiuser Audio/Video Applications. Proc. 2nd Int'l. Workshop on Network and Operating System Support for Digital Audio and Video, Heidelberg, Germany, November 1991.

Broderson, R. 1994. The Infopad Project's Home Page. World-Wide Web Page, http://infopad.eecs.berkeley.edu/.

Chaffee, G. 1994. Personal communication.

S. Gibbs, S. et.al. 1991. A Programming Environment for Multimedia Applications. Proc. 2nd Int'l. Workshop on Network and Operating System Support for Digital Audio and Video, Heidelberg, Germany.

Hamakawa, R. et. al. 1992. Audio and VideoExtensions to Graphical User Interface Toolkits. Proc. 3rd Int'l. Workshop on Network and Operating System Support for Digital Audio and Video, San Diego, CA.

Hewlett-Packard, IBM, and Sunsoft 1993. Multimedia Systems Services (Version 1.0), response to Multimedia System Services Request for Technology. Interactive Multimedia Association.

Ho, S. 1994. Personal communication.

Jayant, N., Johnston , J. , and Safranek, R. 1993. Signal Compression Based on Models of Human Perception. Proc. of the IEEE, Vol 81, No. 10, p1385-1422.

Koegel, J.F., et.al. 1993. HyOctane: A HyTime Engine for an MMIS. Proc. ACM Multimedia 93, Anaheim, CA.

Lee, R. 1994. Personal communication.

Rossum, van G., et.al. 1993. CMIFed: A Presentation Environment for Portable Hypermedia Documents. Proc. ACM Multimedia 93, Anaheim, CA.

Rowe, L.A., and Smith, B.C. 1992. A Continuous Media Player. Proc. 3rd Int'l. Workshop on Network and Operating System Support for Digital Audio and Video, San Diego, CA.

Rowe, L.A., Patel, K., and Smith, B.C. 1993. Performance of a Software MPEG Video Decoder. Proc. ACM Multimedia 93, Anaheim, CA.

Smith, B.C. 1994. Fast Software Processing of Motion JPEG Video. Proc. ACM Multimedia 94, San Francisco, CA.

Steinmetz, R., and Fritzsche, J.C. 1991. Abstractions for Continuous-Media Programming. Proc. 2nd Int'l. Workshop on Network and Operating System Support for Digital Audio and Video, Heidelberg, Germany.

Trehan, R. , et.al. 1993. Toolkit for Shared Hypermedia on a Distributed Object Oriented Architecture. Proc. ACM Multimedia 93, Anaheim, CA.

C Multimedia System Support and Abstractions

C1 The Pegasus Approach to Multimedia

Sape J. Mullender, University of Twente, The Netherlands

C1.1 True Multimedia

Early computers just processed numbers. Soon afterwards, computers also began to be used for processing text, and, today, graphs and images can also be processed. Information can be expressed in the media of numbers, texts, graphs and images and we could call systems *multimedia systems* when they are capable of processing information expressed in any of these media in arbitrarily complex ways.

Usually, however, the term *multimedia* is reserved for the combination of the ‘static' media of text, graphics and images with the ‘dynamic' media of audio and video. But most systems advertised today as having ‘full multimedia capabilities' do not provide any serious capability for processing information presented as audio or video. All they can do is copy audio and video from one place (e.g., camera or CD-ROM) to another (e.g., display). I do not like to use the predicate ‘multimedia' for such systems.

True multimedia must include the capability for programmers and users to process the *information* encapsulated in all the different media. The fact that we can make computers do a better job of searching for names of people in a textual database than searching for their faces in a video database is not the point; the point is that the structure of a multimedia system should be such that, once we know algorithms for processing information contained in video streams, we can sit down and write the programs.

The support of audio and video requires systems to have a strong notion of *timing* – of performing not only the correct actions in the correct order, but also at the correct time. Multimedia systems share this property with *real-time systems*, but the two systems are not the same. Real-time systems have to provide hard guarantees that all deadlines will be met and this is only possible when the load on the system is bounded *a priori*. Multimedia systems will have no such bounds, so they must be designed to do a reasonable job under overload situations. Managing overload is commonly referred to as making *quality-of-service* (QoS) guarantees.

In the next sections, I will describe the architecture of an operating system designed for the support of multimedia applications which is being developed in the Pegasus project, an ESPRIT basic research project[1] of the Universities of Cambridge and Twente. An overview of our systems work was recently published by Mullender, Leslie and McAuley [1].

C1.2 Hardware Support for Audio and Video

Support for audio and video will obviously be provided by a combination of hardware and software. The increased speed of today's processors and size of their memories makes it possible to do work in software now that could previously only be done by dedicated hardware. The advantage of using software rather than hardware is increased flexibility and decreased cost. One can argue, therefore, that it is better to do as much of multimedia processing in software as is possible.

In this section, I will look at the hardware support needed for capture, rendering, and communication of audio and video and explain the Pegasus approach.

For capture of video, one needs a camera and hardware to digitize the video signal. Typically, it is then represented as a sequence of matrices of pixels, one matrix per frame. Most of the video-capture hardware on the market today are boards that interface to a workstation's (or a PC's) bus, and frames are transferred from memory on the capture board to the workstation's memory for processing. This transfer is often slow and only a few frames per second can be transferred in this way. Naturally, this severely limits the opportunities for processing video in real time.

As machines get faster, this problem is likely to go away --- the bandwidth necessary for high-quality video is limited by the needs of the human eye so it will not increase by much. For most platforms, however, the problem still very much exists, so they combine video capture and rendering in one hardware device which is capable of mixing digitized video with the pixels of the window system. The window-management software can specify exactly where the video goes, so video windows can be located anywhere on screen.

This setup should be viewed as a temporary one to overcome the speed mismatch of workstation processors and buses on the one hand and video data rates on the other. Digital Equipment Corporation manufactures a video capture and compression board for their Turbochannel Alpha processors now that only captures frames and stores them in memory. The processor is fast enough to display the frames using the X11-server. Rob Pike constructed a new window system for Plan 9 [2] that can render video well on powerful PCs.

1. Esprit BRA project 6586, September 1992 -- September 1995

Audio I/O is often combined on one device centered around a digital signal-processing (DSP) chip. The data rates are low enough for software to keep up.

For the transmission of video and audio, networks are needed with sufficient bandwidth and low-enough latency. Ethernet fulfils neither requirement. ATM networks are networks with small packets (for good latency characteristics) and often high capacity (for good throughput) and these are eminently suitable for multimedia data traffic. To what extent bandwidth-reservation algorithms should be used is still a matter of much heated debate. For local networks, which usually have a higher capacity than public networks, bandwidth reservation is likely not needed.

Current applications still do little video processing apart from video compression and decompression. Only very powerful processors can do full-frame-size compression and decompression in real time. Compression hardware exists and it is likely that, at least for some time, this will be used.

C1.3 Networked Devices

In the Pegasus project, video-capture and -rendering devices are directly connected to the ATM network [3]. The *ATM camera* [4], in principle, reads out and digitizes the camera's CCD chip, optionally compresses the pixels, wraps up pixels in AAL5 frames, and sends them out into the network. (The prototype implementations actually digitize PAL output from an ordinary video camera.)

A workstation controlling an ATM camera can thus have video sent to another workstation without any data having to pass the workstation's bus. It has been argued that, in very few years, workstations will be powerful enough to handle the video, so a video card in the workstation can do the job too. This is undoubtedly true, but the ATM camera maintains the advantage that video does not *have* to be processed by the workstation. In addition, it has been our experience that ATM devices are simple to construct. An ATM network interface is not really more complicated than the interface to a modern bus. What's more, there are far fewer different ATM interfaces than there are bus interfaces. All in all, we believe that ATM cameras are a big win and, for many of the same reasons, we are also building ATM audio devices.

Although it is quite possible to use the display of a powerful workstation, assisted by appropriate software, to show several video windows simultaneously (at the DEC Systems Research Center I have seen four full-size live video windows on a single Alpha display), we have also built a networked display device where different input channels (ATM virtual circuits) are connected to different areas on the screen (windows). Again, we believe we demonstrated that, with very simple hardware, it is possible to build very good support for display management and video-in-a-window.

With hardware of roughly the same complexity as in conventional workstations, but arranged in a different architecture, we are able to off-load a considerable amount of work from those resources in a system that tend to be most stressed: CPU, bus and memory bandwidth.

C1.4 Scheduling

Applications that process continuous media, such as audio and video, interactively must do so with very low latency. Human interaction is very sensitive to timing (anyone who makes telephone calls via a satellite connection becomes acutely aware of this) and it is the job of good multimedia software, combined with good multimedia process scheduling to make sure that latencies are kept low.

Almost all interactive continuous-media processing involves operations on every audio sample or video frame; that is operations of each application are carried out with frequencies of 25 Hz or higher, resulting in hundreds of rescheduling operations per second. This leaves very little time to make individual scheduling decisions whenever a multimedia process blocks.

In Pegasus, we are investigating a scheduling technique that is used in critical real-time systems such as MARS [5]. Whenever an application is started or stopped that requires periodic scheduling at continuous media rates, we compute a schedule that is presented to the operating system as a table that describes, for each tick of the clock, which threads to wake up. When the end of the table is reached, the operating system wraps around back to the beginning.

In this scheduling, we have to deal with a complication that safety-critical real-time systems are not likely to have: threads can misbehave and use more processor time than agreed on.This can cause other threads to miss their period once. The operating system makes sure of this by terminating processes with unreliable timing. Well-behaved processes must be prepared anyway to miss a period every now and then as a consequence of network delays or transmission errors. The loss of a single video frame or audio sample is not normally noticed by the human observer.

C1.5 Integration into Operating Systems

The Pegasus architecture consists of multimedia workstations (conventional workstations with ATM cameras and ATM display), a storage server that gives bandwidth guarantees for continuous-media data and conventional workstations.

On the machines that require timing guarantees (i.e., multimedia schedulers) --- the multimedia workstations and the multimedia storage server --- we run a tiny operating-system kernel called Nemesis. Nemesis currently provides no real compatibility with existing operating systems (i.e., Unix). Instead, we split our multimedia applications into a timing-dependent part and a Unix-dependent part. The timing-dependent part runs on Nemesis, the Unix-dependent part on Unix.

This set-up is not intended as for possible future production systems. There, we expect that the functionality of Nemesis will be incorporated into a Unix kernel of some kind. The current set-up does, however, provide us with an environment in which we can experiment easily and measure performance without being hindered by peculiarities of existing operating systems.

Acknowledgements

The work presented here is the work of a team whose members we gratefully acknowledge: Paul Barham, Ralph Beckett, Richard Black, Peter Bosch, Tatjana Burkow, Shaw Chaung, Simon Crosby, Feico Dillema, Richard Earnshaw, Dave Evers, Robin Fairbairns, Daniel Gordon, Mark Hayter, Paul Havinga, Eoin Hyden, Pierre Jansen, IanLeslie, George Linnenbank, DerekMcAuley, George Neville Neil, Ian Pratt, Timothy Roscoe, Paul Sijben, Gerard Smit, Martijn van der Valk, Lars Vognild, and Lex Warners.

References

[1] S. J. Mullender, I. M. Leslie and D. R. McAuley, Operating-System Support for Distributed Multimedia, Proceedings of the Summer Usenix Conference, Boston, MA, June 1994

[2] D. Pressetto. R. Pike, K. Thompson and H. Trickey, Plan 9, A Distributed System, Proceedings of the Spring 1991 EurOpen Conference, Tromso, Norway, May 1991, 43-50

[3] I.M. Leslie and D.R. McAuley, Faiisle: An ATM Network for the Local Area, ACM Computer Communication Review 21(4), September 1991

[4] I. Pratt, ATM camera V1, in ATM Document Collection 2 (The Orange Book), University of Cambridge Computer Laboratory, New Museums Site, Pembroke Street, Cambridge, CN2 3QG England, Feb. 1993, 28, 1-28,7

[5] H. Kopetz, A. Damm, C. Koza, M. Mulazzani, W. Schwabl, C. Senft and R. Zainlinger, Distributed Fault-Tolerant Real-Time Systems: The Mars Approach, IEEE Micro, February 1989, 25-41

C2 Interactive Multimedia Services on Cable TV Networks

P. Venkat Rangan, University of California at San Diego, USA

A recent problem in our research on Video on Demand has been the economic viability. Economic viability mandates the maximum number of subscribers being serviced with the existing infrastructure. In this paper we propose information caching as the key to make Video on Demand lucrative. In this paper we examine the outstanding research issues in the context of implementation of Video on Demand on cable TV networks.

C2.1 Motivation and Objective

The *leitmotif* of our research in Personalized Video on Demand has been to amortize the enormous network and storage costs. This is supported by a recent survey [2] which showed that while the personalized Video on Demand has a high appeal among the 44% of the respondents who were willing to pay for it, only a minuscule 14% were willing to pay *more* than the existing cable rates for it. Nearly two thirds of the respondents owned personal computers with nearly half of them equipped with a modem; hence we can infer that they knew pretty well what a difference, Video on Demand would make in their lives. Hence amortization of network and storage costs and the consequent reduction in the per user costs is a factor which plays a crucial role in the success of personalized video on demand services.

In this paper, we first describe the types of interactive services and we differentiate them on the basis of the extent of control they give the user. Next, in Section C.2.3, we outline the current cable TV architecture and the expected enhancements in the future. In Section C.2.4, we propose information caching as a key to amortizing network and storage costs. In the next three sections, we describe the types of viewer requests and the issues we face in servicing them. Finally, we contrast cable and telephone network architectures and their suitability for true video on demand.

C2.2 Types of Interactive Services

Interactive services can be broadly categorized into three classes based on the extent of control they give the viewer:

Pay-per-View (PPV) now extant, enables cable subscribers with set top decoder boxes to order movies and programs that they might have missed, at specific announced times. But the set top boxes do not give the viewer any freedom except that the viewer has the choice whether to view the program or not and (s)he pays for viewing the whole movie if he pays at all. Neither does the viewer have any temporal control over the movie shown.

Near Video on Demand (N-VOD) is achieved by having many channels broadcast the same program but with a definite temporal variation or delay between the channels so that the viewer has simulated forward and reverse functions by changing channels appropriately. A popular suggestion is to have movies broadcast with a start time delay of 15 minutes so that the viewer experiences a rudimentary form of simulated forward and reverse functions by switching channels and also, the viewer has some control over the time at which (s)he can view it. This implies that for a one and a half hour movie, six channels are taken up for N-VOD. For live programs, deferred airing has been suggested to give the user a certain amount of choice in deciding the time at which he should view the program. Deferred airing is a concept in which live programs are broadcast after a specific time delay so that viewers can see it in case they have missed the liveprogram.

True Video on Demand (T-VOD), in contrast to all of the above, subsumes the functions of a VCR and hence gives the user, complete freedom to alter the viewing temporally. For T-VOD, multiple channels are rendered unnecessary but a switching system needs to be installed to support bi-directional signalling.

The first two do not need much enhancement in the cable network architecture. The implementation of the third one brings up many issues yet to be resolved.

In the next section, we examine the cable network architecture and its suitability for true video on demand.

C2.3 Cable Network Architecture

In this section, we examine the architecture of the existing cable network and outline the expected enhancements in it. The cable network in its present state of development is architecturally unsuitable for the implementation of true video on demand. The architectural enhancements that we have outlined here make true video on demand feasible on cable networks.

C2.3.1 The Existing CATV Architecture

The existing cable network architecture, shown in Fig 1, is a tree structure having the following basic components:

- *Headend* is the location of the switching equipment that collects and transmits programs.
- *Trunk cables* are the co-axial cables that transmit signals from the headends to the regional nodes. Losses in the trunk cable are made up by trunk amplifiers, which reduce the degradation of the signal due to noise and attenuation.
- *Feeder cables* are Coaxial cables that run from the amplifiers on the trunk cables and serve as buses from which users can be connected to the network. Taps, which are directional couplers, are placed periodically along and in series with the feeder cable for providing service connection point for drops. Line extenders decrease the losses in the feeder cables.
- *Drop cables* are coaxial cables connecting the user to the nearest tap.
- *Express feeder cables* are designed to minimize losses as shown in the Figure 1. No taps are on the feeder cables but separate cables that feed forward and backward contain the taps.

The main drawbacks of the above existing architecture that are, nevertheless expected to be improved in the near future are:

- *Bandwidth Crunch*: The existing bandwidth on the cable networks is insufficient to withstand peak hour network load.
- *Signalling* needs to be changed to digital to enable compression and efficient switching like the ATM.
- *Broadcast transmission*: No switching and hence all signals broadcast by the main station are received by all the subscribers.

C2.3.2 The Future CATV Architecture

The expected cable network of the future, shown in Fig 2, is expected to have significant enhancements. The coaxial cable to the headends will be replaced by optical fiber cables and the headends will be connected in a ring pattern to the regional hub. So, the pattern will still retain a logical tree, though topologically it will be a hybrid of the ring and star. Each regional node passes around 500 homes and each headend passes around 10000 homes. We envisage one principal storage provider which serves as a database for thousands of movies and is at the highest level in the hierarchy of the cable network. Throughout this paper, we refer to it as the *main storage provider.*

The expected architectural enhancements are listed below.

- *Fiber Optic cables*: The network of the future will have high bandwidth fiber optic cables running from the main storage provider to the headends.

- *Compression*: Compression increases the no of channels that can be transmitted tenfold and gives flexibility to the subscriber either to use HDTV or NTSC.
- *Switching*: An efficient switching mechanism like the ATM which couples the advantages of both circuit and packet switched networks and installation of photonic switches are essential.

In the next section, we examine how efficient use of bandwidth can be made so as to achieve a tradeoff between servicing the maximum number users and minimizing the cost per user.

C2.4 Information Caching – An Answer to Undesirable Peaks?

Studies [1,3] have shown that the patterns of viewership are very much dependent on the time of the day. Akin to the telephone network, there are peak hours during which the traffic is maximum and the majority of the titles demanded is a small subset of the most recent set of hit movies. This has sparked off considerable debate and confusion whether True Video on Demand is more lucrative than Near Video on Demand. Caution has been expressed [3] about the success of True Video on Demand because it is felt that Cable and Telephone companies pushing Video on Demand are investing heavily to create massive storage providers to deliver thousands of programs when all that consumers want are the eight or ten hit movies very month.

Information Caching has the potential not only to smoothen out the peak, but also to amortize storage costs and maximize network utilization. In the system configuration we propose, a synergy between several enterprises occurs: *Storage Providers* that manage information storage at multimedia servers (a role akin to that of video rental stores and libraries in today's context), *Network Providers*, that are responsible for media transport over integrated networks (a role akin to that of cable and telephone and cable companies of today), and *Content Providers* such as entertainment houses, new producers etc., that offer a multitude of services to subscriber homes using multimedia servers and broadband networks. In the proposed architecture, multimedia servers of suitable capacities are installed at the different hierarchical levels and function as temporary caches for information delivered from metropolitan repositories. The locations of switching equipment in the network are obvious choices for locating the multimedia servers. For managing distribution of the selected multimedia programs to each subscriber's home, we propose *Distribution Agents* hat negotiate the storage and retrieval times of programs with storage and network providers, and program authorizations with the content providers. In this process, the distribution agents judiciously determine locations of programs and their distribution times depending upon the anticipated demand for the future, the costs of storage and the utilization of bandwidth.

To facilitate the process of program selection, we propose *Personal Service Agents* (PSAs) that search and locate programs that match each subscriber's needs, and customize the content providers' offerings to deliver personalized services [5]. The main functions of a PSA are: (i) It monitors each subscriber and constructs a behavioral profile for the subscriber. (ii) Based on this profile, the PSA queries the network directory service to identify content providers offering relevant programs. (iii) Then, the PSA contacts each of the content providers, navigates through their content bases, and constructs a list of programs relevant to the subscriber. (iv) Once the subscriber makes a selection of programs to view, the PSA continuously interacts with the content bases and chooses portions of programs and even their levels of detail to display to the subscriber. (v) The PSA also solicits explicit relevance feedback from the subscriber for its selections, so as to tailor its selections in the future to better suit the subscriber's needs [9].

In the next two sections, we develop a mathematical model of the cable network and efficient techniques for information caching on the cable networks.

C2.5 Cable Networks and Tractability

The mathematical model of the cable network and its tractability are outlined in this section.

Assuming that there are N nodes in total, we define two $N * N$ matrices. One is the maximum bandwidth matrix M and the other is the available bandwidth matrix A. The entry in *row i and column j* in the matrix M, $m(i,j)$ gives the maximum available bandwidth on the link from the *node i to node j*. Note that $m(i,j)$ and $m(j,i)$ denote the upstream and downstream bandwidth between the *nodes i and j* and they need not be equal. Similarly $a(i,j)$ and $a(j,i)$ denote the free bandwidth available for transmission upstream and downstream between *the nodes i and j*.

The elements of the matrix A are functions of the time of the day and the corresponding elements of the *matrix M*. The elements of the *matrix M* are a function of the level of hierarchy at which the *nodes i and j* are located. This dependency of the maximum network bandwidth on the level of hierarchy at which it occurs, is because, the headends, at the higher level, are connected with each other by an optical fiber ring which has a high bandwidth, while, the drop cable to the user homes, at the lower level are the copper co-axial cables which are lower in bandwidth. So, the maximum bandwidth decreases down the hierarchy.

Coming to the storage providers, the maximum storage capacities of the storage providers at the nodes, is assumed to decrease as we go down the hierarchy. This is reasonable assumption because the storage requirements are proportional to the number of users served at each node. This number decreases as we go down the hierarchy. The space available for storage at a node is a function of two factors. One is the maximum bandwidth and the other is the time of the day. The bandwidth usage peaks during specific hours, for example, in the evenings when everyone returns from work and is the time for relaxation. This is a generalized raw pattern of the traffic applicable to any of the nodes or any of the links between nodes. This is derived from the demand pattern of the user at the lowest level. This demand pattern induces a similar usage of bandwidth and storage. The problem we face is that the peak traffic is greater than the maximum possible utilization of the bandwidth and storage space.

The solution we need, is the optimal alternating sequence of the storage and transmission times. In an optimal solution, we endeavour to smoothen out the peaks for individual links so that the peak lies utilization well below the maximum possible utilization.

We approximate the usage pattern of bandwidth and storage, to a Gaussian distribution. This constitutes a set of assumptions. The times at which user requests need to be serviced, the assumed traffic pattern and the storage and bandwidth constraints at each of the nodes constitute a set of constraints.

So, the above assumptions and constraints yield a solution for the optimal caching strategy of the first delay tolerant request. This solution for the first request by itself is an effort to smoothen out the peak. Hence, the set of assumptions for determining the optimal caching strategy for the second delay tolerant request have to take the caching strategy for the first request into account because it alters the traffic pattern. This yields another optimal solution and hence alters the assumptions for solving the third request. So, the assumptions for the *k*th request take into account the alterations in the traffic pattern due to the solutions for the *(k-1)* requests considered prior to this. So the globally optimal solution is the result of a recursively computed optimal solution for each request.

This problem can be shown to be NP complete [7]. In the next section, we propose a caching strategy which follows the old adage *Prevention is better than cure*. The *Anticipatory Caching Algorithm* that we propose anticipates the movies or programs that consume maximum bandwidth and use *information caching* to service them at the lower levels. This smoothens out the peak at the higher levels because the algorithm effectively prevents the user request pattern from getting translated into bandwidth and storage usage pattern.

C2.6 The Anticipatory Caching Algorithm

Success of true video on demand is in jeopardy due to the peak hours in the viewership pattern. The caching algorithm that we present in this section, has the potential to convert this seemingly discouraging viewership pattern into an advantage maximizing the utilization of the available bandwidth and hence maximizing the number of requests serviced.

An important factor desired in the video on demand on the cable network is ubiquity because the more ubiquitous the cable network, more the geographical area covered by the system, given the common costs borne by all the subscribers such as equipment costs for switching and storage, the lower the average cost per user served. This is because the bigger the infrastructure, the lower the unit cost of development. The cable headends and the regional nodes are distributed over a wide geographical area spanning different time zones. (For example, the time difference between the east and west coasts in USA alone is 3 hours). But such large areas have only one main storage provider as the storage costs are enormous. The Anticipatory Caching Algorithm takes advantage of the fact that only one main storage provider serves large areas with significant time differences.

C2.6.1 Assumptions and Implications

Let us assume there are 10 hit titles which account for 80% of the movies demanded in the peak hours, say from 6:00pm to 10:00pm. Surveys have shown that this is a very reasonable assumption.

Consider the hierarchy of the cable network as based on the division according to time zones. The hierarchy is actually based on the number of users served. The number of users served translates into geographical distances which in turn may translate into differences in the time zones. So, the assumption we made is a reasonable one.

Though we have assumed bidirectional flow of data below, the present CATV architecture needs a technological enhancement to permit the switching and control of bidirectional signalling.

Further, we propose that each node in the lowest level of the hierarchy,i.e., the level just above the user, should have a storage provider. By saying this, we do not imply that each storage provider should have a capacity equal to that of the main storage provider which is extremely large. Logically, the maximum available storage space should decrease as we go down the hierarchy. Having storage providers at every node is purely for the purposes of information caching. These storage providers can either be owned by the cable company or the storage space can be rented on a per hour basis. A MPEG compressed movie of about 90 minute duration occupies roughly 1 Gbyte of storage space. The storage providers at each node in the lowest level of hierarchy having a storage space of around 10 Gbytes, in other words, a storage space for ten digitized movies, will prove to be a key investment and any extra expenditure incurred due to this is offset by the large amount of network bandwidth which gets freed up, relieving congestion, as outlined below.

C2.6.2 Caching by Anticipation

The heuristics of the Anticipatory Caching Algorithm for delivery to multiple neighborhoods are described here. The fundamental principle behind this algorithm is that the 10 hit titles are cached at the nodes in the lowest level of hierarchy during the peak hour irrespective of how many requests are received and for which titles they are received.

Each node is assumed to have a storage provider. The primal node (the one which falls in the first time zone) has no option but to get the 10 hit titles from the main storage provider and cache it for the 4 hours from 6:00pm to 10:00pm.

The PSAs (Personal Service Agents: see Section 4) which are at the lowest level of hierarchy should have a list of the nodes which are its immediate predecessors and successors with respect to the time zones. So, the PSA at the node next to the primal node will get the 10 hit titles from the primal node at the start of the peak hour and cache all the 10 titles for the next 4 hours.

Each PSA has the task of comparing the two options namely getting the 10 titles from the main storage provider or getting them from its immediate predecessor storage provider. The options have to be compared on the basis of the minimum bandwidth available on the two paths and the time delay induced in the paths due to switching at the nodes enroute.

For example, if the immediate predecessor happens to be at double the number of hops compared to the main storage server, there is no point getting the data from the predecessor as it defeats the very purpose for which it was used.

Many storage providers co-exist in the same time zone, and during the peak hours, all of them would have cached the 10 titles. The immediate successors will be faced with the problem of deciding which predecessor they have to get the titles from. To resolve the dilemma faced by the PSA in such cases, we propose a greedy algorithm in which the PSA will choose to get the data through the line which has the maximum available bandwidth for transmission.

In the absence of the anticipatory caching strategy, assuming that all the titles requested are obtained by each PSA directly from the main storage provider, more than 80% of the bandwidth right from the first level to the last level is used up for these 10 titles for at least a time period of (T(max) + 4) hours where T(max) is the maximum time difference between any two nodes at the lowest level of hierarchy.

In contrast, using the Anticipatory Caching Strategy, only at the lowest level, $80\%$of the bandwidth is used up for an average time period of 4 hours for these 10 titles. So, the rest of the bandwidth in the majority of the lines and higher levels of hierarchy is freed up for serving clientele with other preferences during the peak hours which are the key hours determining the success or failure of the caching strategy.

An enhancement in the cable architecture which gives our caching algorithm a considerable edge over the other strategies is the direct connection of the headends with opticfiber cables. This architecture enables the cached data to be transferred directly between two adjacent headends without the data going to any other higher levels of hierarchy. The existing architecture mandates the cached data to go all the way from the source headend up to the main storage provider and bounce back all the way down to the destination headend.

In the next section, we outline the architectural differences between the cable and telephone networks and analyze the impact of these differences on the implementation of T-VOD.

C2.7 Cable Network vs. Telephone Network

Finally, in this section, we contrast the two fundamentally different, prospective networks for the implementation of T-VOD and examine their architectural suitability to support true video on demand on a large scale.

C2.7.1 Architectural Viewpoint - The Bandwidth Crunch

Ever since the FCC favored the establishment of the *video dial tone*, the telephone companies have been trying to develop and test the technology for T-VOD. Let us examine the architectural suitability of the telephone network for T-VOD. The telephone network has a hierarchical structure which is suitable for a central storage provider serving the requests of any of the PSAs at the nodes.

The greatest technological advantage that the telephone network has is the efficient switching mechanism which allows bi-directional signalling between any two nodes. This gives a high degree of control over the flow of information. So, there is not much further capital investment in the form of switching equipment that is necessary for implementing the T-VOD over the telephone network.

But, the greatest drawback is the bandwidth crunch. The copper wire of the telephone network cannot withstand more than 1.5 Mbps which yields not more than one inferior quality video channel. So the low investment cost advantage is more than offset by a failure to meet the QoS requirements of T-VOD.

The co-axial cable network on the other hand has enormous bandwidth that is expected to reach 1GHz, translates into 80 5-Mbps digitally compressed channels compared with the inferior one 1.5 Mbps channel available over the telephone network.

C2.7.2 Economic Viewpoint - The User Perspective

The essential difference between the billing in the cable network and the telephone network is that the telephone network charges the user for the time for which the network is used while the cable network charges only for a service provided. To elucidate the difference, consider the difference in the billing system between the two networks.

If T-VOD is achieved on the telephone network, assuming storage providers are available for caching, the users are charged proportionately for the duration for which the multimedia programs are cached at intermediate storage providers and for the duration of a live connection i.e. the charge is directly proportional to the time for which the telephone network is used for transmitting the movie. So, if the viewer alters the rate of viewing, the charge varies proportionately. This is a major drawback because the viewers would undoubtedly find the simple home video rental system to be much more lucrative because they have the freedom to reverse, forward and view it more than once for the a much lesser cost. Thus, the telephone network with the existing billing system would fail economically for the implementation of T-VOD.

On the other hand, the existing cable billing system with the exception of PPV, charges a flat rate for the user depending on the number of channels. The absence of a switching system leaves the cable operators with no other option. But for implementation of T-VOD, this billing system would prove absurd because the viewer using the network most of the time would be charged the same rate as a viewer who does not view anything at all! This would be unreasonable. Hence, we propose that the charges for T-VOD on the cable network should be akin to the existing PPV charges which is roughly the same as home video rental charges. It is up to the cable companies to minimize the expenditure they incur by employing efficient caching techniques which would maximize the utilization of the network bandwidth and hence maximize the number of simultaneous requests serviced.

C2.8 Closing Remarks and Further Work

The key factor in the success of True Video on Demand is that the network should have the necessary infrastructure to withstand a sudden upsurge of network traffic and bandwidth usage. Each node at the lowest level of hierarchy passes 500 homes and each headend which is just on top supports around 20 such nodes or 10000 homes and each hub node which is just on top of the headend supports tens of thousands of users. True Video on Demand, if implemented on such a complex and massive architecture is bound to collapse if all the requests are serviced at a higher level in the hierarchy, during peak hour.

In this paper, we have presented caching algorithms that smoothen out the peaks and hence make the network capable of withstanding the upsurge in user requests during the peak hour.

Work yet needs to be done, to determine a near optimal caching strategy for cable networks based on the mathematical model as described in Section 5. As the problem is NP complete, an approximate caching strategy needs to be determined and compared with the optimal one. The number of parameters to be considered in determining a near optimal caching strategy for cable networks is much more than that for telephone networks and hence it poses a very challenging problem.

Acknowledgements

We are grateful to Prof. Christos Papadimitrou for the very fruitful discussions we had with him.

References

[1] Cable Television Laboratories Inc., Digital Media Servers, *Request for Information*, 1994.
[2] H.A. Jessel, Cable ready: The high appeal of interactive services, *Broadcasting & Cable*, May 23, 1994.
[3] H.A. Jessel, Sie pessimistic over video on demand, *Broadcasting & Cable*, May 9, 1994.
[4] B.A. Kaplan, Cable Television, Communacopia: *A Digital Communication Bounty, Goldman Sachs*, July 1992.
[5] A.Lippman and W.Bender, *News and Movies in the 50 Megabit Living Room, IEEE Globecom*, pages 50.1.1-50.1.6, 1987.
[6] R.B. Morris, Telecommunication Services, Communacopia: *A Digital Communication Bounty, Goldman Sachs*, July 1992.
[7] C.H.Papadimitrou, *Personal Communication*
[8] C.H.Papadimitrou, Srinivas Ramanathan and P.Venkat Rangan, Information Caching for delivery of Personalized Video Program on Home Entertainment Channels, *Proceedings of IEEE International Conference on Multimedia Computing and Systems, Boston*, May 1994.
[9] Srinivas Ramanathan and P.Venkat Rangan, System Architectures for Personalized Multimedia Services, *IEEE Multimedia*, 1(1):37-46, February 1994.
[10] P.Venkat Rangan, System for Efficient Delivery of Multimedia Information, *US Patent pending, San Diego, CA*, January 1994.

C3 Multimedia System Support and Abstractions

Doug Shepherd, Lancaster University, Great Britain

C3.1 Introduction

First, let us say what is our interpretation of "multimedia system support and abstractions". For the purposes of this note, we are mainly concerned with communications, operating systems and distributed systems platform issues and do not include such areas as languages, databases, GUIs, media encodings and document structures. This is not to say that these areas are not important. Clearly they are. However, it seems to us more worthwhile to use the limited space available to go deeper into a more limited set of issues. Furthermore, even though communications are on our list, we don't want to go too much into multiservice network issues. Most of the communications issues we will discuss are end-system communications issues or at least those networking issues that are multimedia specific and not general multiservice network issues (e.g. network management, pricing issues, ...). It will not be surprising to discover that the issues we choose to discuss are primarily those under investigation at Lancaster!

C3.2 What We Know

Networking and communications support for multimedia is quite well developed. We know (or almost know) how to support guaranteed and best effort traffic in networks. At least on a point to point basis. QoS parameters of relevance are generally agreed to be bandwidth, latency and error rates but not jitter. Important recent developments include fast LANs and ATM. RSVP looks like it will fulfil most of the requirements of QoS in the Internet (although IP routers themselves do not yet sufficiently support QoS). There is also IP multicast which is a useful step forward which, however, fails to accommodate QoS. It is a dissemination oriented service rather than a 'connection oriented' multicast service. Also, there has been significant work on transport protocols. These tend to be minimal in functionality for continuous media but they are still necessary to provide port-to-port addressing, selective error management and transparency over heterogeneous networks.

In the area of workstations, the latest commercial machines are capable of supporting some level of multimedia in terms of raw speed. It is mainly system software that is the problem (see below). However, there is also an architectural problem with current workstation which arises from the inherently non scaleable bus based interconnect used in most of these machines. To address this limitation, novel multimedia workstation designs based on message passing architectures have been explored - e.g. the DAN at Cambridge and the LANC/MEND at Lancaster. These designs exploit a cell switching interconnect inside the end system as well as in the network and are structured as loose confederations of processors and devices each of which is geared towards supporting a single medium. Such architectures are more scaleable than current workstations both in terms of the number of processors supportable and the scalability of the interconnect.

The OS field is less developed. Although it is possible to run multimedia applications under standard OSs such as UNIX, there are limits which can very easily be reached. The main problem is that UNIX has been shown to be particularly fragile when running multimedia applications. An unfortunate application mix can destroy all the required timing properties of continuous media in an unpredictable and uncontrollable way due to the use of scheduling algorithms geared to fairness and responsiveness rather than predicatbility.

One system support service that has received a lot of attention in research circles is multimedia synchronisation. It is now known what the requirements are in this area. Roughly, supporting inter- and intra-stream synchronization where the to-be-synchronized streams may originate on different source hosts. Real-time event-based synchronization must also be supported. Much work has been done on abstractions for expressing these requirements and some work has been done on mechanisms and protocols to realize the necessary underlying mechanism. However, as yet little real experience is available.

C3.3 Remaining Problems and Research Issues

As stated above, the majority of the problems we want to discuss are end-system problems. However, first we will mention a number of multimedia networking issues. The biggest of these is the provision of multicast services. Such multicast services as there currently are do not support QoS. There are two main issues here: receiver heterogeneity and network QoS support. Attempts to address the first problem are currently exploring the concept of filters in the network or end-system which can adapt a canonical sender determined QoS to the levels supportable by particular receivers. The answer to the second problem is some form of QoS based routing or policy based routing. Little work has yet been done in this area - either in the case of multicast or point to point communications for that matter. As mentioned above, the interaction of multimedia synchronization and multicast must also be investigated. In the most complex scenario, it should be possible to ensure inter-stream synchronization on all receiving end-systems where the multiple sources are on separate hosts and are communicating with their (heterogeneous) receivers over multicast flows. It is not clear that this scenario is actually a real requirement but if it turns out to be it is certainly not yet known how to achieve it!

Returning to the end-system and specifically the topic of operating system support, there are a number of outstanding issues. First, in terms of thread scheduling, some researchers believe that the necessary real-time characteristics can be achieved simply by structuring applications as adaptive applications (i.e. applications which monitor fluctuations in QoS and adapt their way of working when it changes). There are some existing applications (e.g. vat, ...) structured along these lines which seem to work well. Others believe that more support is needed from the OS. In particular, it is argued that more predictable scheduling policies are needed which offer sufficiently graceful degradation modes that applications have a chance of adapting to. It is also argued that means of re-negotiating the QoS of a multimedia session are necessary to allow the user to reassess their options when, say, starting up a new application and realizing that already running applications can be downgraded to give improved service to the new one. Finally, it is important that these hypothetical new scheduling policies are able to co-exist with non-real-time scheduling policies so that 'ordinary' applications can run in the same workstation environment. There is a severe lack of experience in all these areas. Real systems need to be built and evaluated. More adaptive applications need to be written and tried out. The right balance between 'hard' and 'soft' real-time performance needs to be determined.

In addition to the requirement for supporting real-time streams of continuous media, there is also a requirement to support control messages with bounded latency. The issues raised here are slightly different from those discussed above. In some ways latency bounded control messages are harder to support as you may not be able to afford a temporary 'slippage' or 'glitch' which would often be acceptable in a continuous media stream. Instead, if the system contracts to deliver a message in n milliseconds (thread to thread) it must provide a rather strong guarantee that it will actually do so.

Virtual memory systems also impinge on a system's ability to support time constraints. It is of little use to schedule a thread at just the right time if the first thing the thread does is access a variable which resides in a swapped out page which takes an indeterminate time to load. One extreme solution to this problem is to 'wire' pages when timing constraints must be met. This, however, can quickly become problematical as the reduced number of usable page frames leads to thrashing. Each thread with time constraints must wire its code, stack and buffers which consumes a considerable amount of physical memory. An alternative approach is to introduce the concept of 'QoS controlled memory' whereby pages offer time bounded guarantees on their access latency. Although ideas such as pre-paging and improved swap device data layouts have been put forward, very little research effort has yet been directed at this problem. Another requirement is a flexible set of options for 'touching' or not 'touching the data'. Many applications are content with merely supervising the flow of continuous media data as it passes between the network and a local device. For example, the ability to start and stop the data flow may be all that is required. However, there are increasing demands for applications able to touch the data as it passes so that they can perform on-the-fly manipulation of video or audio. An intermediate requirement is that applications often want to know the exact position of the frames of a media stream as it flows from the network to a device. This information is needed so that applications can synchronize other activities with the flow of data. One way that all these requirements can be met is by insisting that data is piped through an application which sits between the network and the device in a tight loop going: while(1) { get_data(); put_data();}. However, this incurs unnecessary protection boundary crossings and context switches for applications that don't care to see the data. Therefore mechanisms must be provided that allow application designers to make a choice between all three alternatives described above with minimal inconvenience and overhead.

An issue related to scheduling is the question of scheduling/ comms integration. In existing systems there isn't any. Data arriving from the network gets put in a buffer and the 'urgency' of this data has no influence over when the thread waiting for that data gets to be scheduled. This is not a good way of working when we are concerned with time critical data. What seems to be needed is the abstraction of a 'session' which subsumes both the network and the scheduling of threads handling the data going to and arriving from the network. Such a session abstraction could also subsume the setting up of QoS controlled memory for the sending and receiving threads as outlined above. Of course, significant mechanism would be required to support such an abstraction. A distributed resource allocation protocol is one obvious requirement.

Speaking of abstractions, it is not clear that the traditional user kernel interface for read() and write() or send() and recv() like calls is the most appropriate for multimedia. Another possibility, first explored by Black in a paper in SOSP almost 10 years ago, is to let the application take on the passive role normally adopted by the system and let the system itself take responsibility for initiating events. This ties in with two points made above: i.e. the need for applications to be made aware of events such as frame arrival even when they don't want to see the data itself, and the need for scheduling/ communications integration. What seems to be needed is a system interface with which applications can register handlers when they establish 'sessions' with a given QoS. These handlers would be upcalled when data is required (at the source) and available (at the sink). Depending on what the application puts in the handler the varying flavours of touching or not touching the data could be achieved (e.g. a handler may be null, may just respond to the events that data is delivered or may actually access the data via a pointer passed in the handler invocation. Such a system initiated interface would ease the scheduling/ communication integration problem by not requiring a synchronization between a protocol and an application thread - the protocol thread can just extend up into application space. An additional attraction of such an inverted interface might be ease of structuring of real-time application code (cf. X windows call-backs).

Another end-system issue which is now receiving some attention is storage services for continuous media. There are now prototypes in existence which are able to deliver of the order of 10's of video streams per unit, but the problems of scalability have not yet been sufficiently addressed. For example, what are the best combinations of replication and striping when deciding how to distributed videos over multiple networked storage servers? How should tertiary storage be managed when its use is necessitated by huge amounts of video material?

Above the OS, there is a growing need for higher level distributed systems platforms to be made multimedia capable. Much progress has been made in recent years in the area of platforms - e.g. ANSA, DCE and CORBA, but none of these systems currently supports multimedia. This is true even at the level of abstractions, never mind underlying mechanisms. For example, most platforms are based on the RPC abstraction for communication and this is manifestly unsuitable for continuous media. Recent versions of the ODP draft standard have attempted to address the requirements of multimedia applications with the introduction of stream interfaces and explicit bindings, but much more needs to be done to develop these ideas even into usable abstractions never mind working systems.

Another issue that comes into focus when we consider distributed systems platforms is the variation between representations of QoS at different system levels. For example, QoS at the network layer is bandwidth, latency and error rates. At the platform level it may more appropriately be frames-per-second, camera-to-screen latency, minimal jitter etc. And there may be a different representation again at the transport level. So, if programmers are to be able to program exclusively at the platform level while being confident of getting optimal service from the lower layers, there must be some means of mapping these varying notions of QoS between the layers. This notion of a layered QoS architecture naturally extends to more than just mapping of representations. There may well be scope for an architecture that encompasses QoS monitoring and adaptation at multiple layers such that each layer attempts to adapt within its means and when it fails it passes the problem on to the next highest layer. On the other hand, this may be too heavyweight a design to realistically adopt in practice. Investigation is required here.

A final area of interest in the end-system arises from the field of mobile computing. There is already talk of sending video over local area radio networks but this, of course, is not currently possible in a wide area context due to bandwidth limitations. Current multimedia mobile systems are limited to text, graphics and voice. However, there are still problems to be solved. For example, how should systems/ applications deal with the ever-present possibilities of disconnection and wild swings in QoS inherent in mobile communications? Can the concept of adaptive applications mentioned above address these problems? Can imaginative pre-fetching and caching policies work for image and audio media? One interesting observation is that, due to the relatively high processing power/communications bandwidth ratio present in mobile end-systems, there seems to be a much greater scope for adaptation to varying QoS (either by applications or systems or both) compared to the situation with end-systems attached to high-speed networks.

C3.4 Research Agenda

The following is a brief summary list of research issues abstracted from the above discussion.

- Investigate feasibility of filters for the support of heterogeneous receivers in multicast communications. Should filters be in the end-system, the network or both?
- Develop schemes for QoS based routing particularly for multicast communications.
- Based on the above, develop multicast services with support for QoS and heterogeneous receivers.
- Evaluate and gain experience with distributed multimedia synchronization mechanisms, protocols and policies.
- Investigate the interaction between distributed synchronization mechanisms and mutilates communications.
- In the OS field, investigate scheduling schemes that degrade gracefully and coexist with standard UNIX-like policies.
- Investigate adaptive QoS management schemes in end-systems (particularly in mobile end systems where QoS fluctuations are large and the processing/ bandwidth ratio is high).
- Investigate the concept of 'QoS controlled memory'.
- Develop resource allocation protocols to implement the notion of an end-to-end session which supports thread-to-thread QoS by subsuming the allocation of CPU, memory and network resources.
- Investigate 'passive application/ active system' based APIs and integrated communications/ scheduling in OSs.
- Develop and evaluate multimedia related ODP concepts in DCE, CORBA etc.
- Investigate the feasibility of a 'QoS architecture' that specifies QoS representation, monitoring and adaptation etc. over a number of architectural layers.
- Improve the scalability of storage services.

D Multimedia Storage and Databases

D1 Multimedia Storage and Databases

Klaus Meyer-Wegener, Technical University of Dresden, Germany

D1.1 Stable Knowledge

There are still some arguments among researchers about what a multimedia DBMS should do or not do. The initial definition was given by Stavros Christodoulakis at the 1985 SIGMOD conference [Chris85]. Other approaches are related with Masunaga [Masu87a], ORION [Woel87a, Woel87b] and VODAK [Klas90a, Klas92a].

In general, the development of extensible and object-oriented DBMSis supposed to offer all the mechanisms needed to manage multimedia data as well as the standard alphanumeric data - an assumption that does not hold yet in my opinion. Hence, the following statements refer to "what I have found out by now" and are not necessarily shared by other researchers.

First, an MMDBS primarily has to play the role of a base system for a variety of different multimedia applications. It is responsible for storage management and retrieval, not for user interface issues. Consequently, the API should be in the focus of interest, not the interactive interface. An end-user interface could be on of the applications that access the DB, and since it has a multimedia interface, it will be a rather complicated piece of software. It should use the API as well and thus have a distinct interface to the DBMS. In summary, an MMDBS may be the heart of a MM information system, but it is not in itself the system.

Second, the various kinds of monomedia data are too different to be mapped to a single storage concept. Text is not the same as video, neither with respect to size nor in the subtle issue of time dependence. BLOBs for instance are too weak a concept to handle those data properly. Allocation of secondary storage as well as buffer management, to name just two mechanisms, have to be adapted to the specific type of monomedia data to deliver the required performance. Media-specific compression techniques give another reason. This does not necessarily mean that the MMDBS has to be aware of the full meaning of "text" and "video" it only means that it should distinguish, let us say, different kinds of BLOBs.

Third, Abstract Data Types must be defined for the monomedia data objects to guarantee data independence. In particular, device independence (e.g. of the various kinds of optical disks) and format independence must be offered for multimedia data. Content addressability, although still an open issue (see below), will certainly lead to different comparison operators applicable to the media data types. Given those ADT's, it is only a secondary question whether they should be embedded into an extended relational or an object-oriented DBMS.

Fourth, the new and rapidly growing world of multimedia network information systems (www, WAIS, gopher, Hyper-G, etc.) is currently incompatible with MMDBS. The reasons are:

- They are not using a DBMS, but only files, so that they have defined very specific data formats that are hard to use by other applications.
- The handling of multimedia is left to external applications, typically viewers (xv, ghostview, etc.).
- They are primarily interactive systems, designed for stand-alone use, not to be integrated into larger applications.

However, one might consider to extend those systems to have them access an MMDBS for data storage.

D1.2 Remaining Problems

D1.2.1 Definition and Standardization of the Media Abstract Data Types (MADT)

The definition of the ADT's is not an easy task. They are far more complex than "point" or "box". Attempts have been made, but they cannot yet claim to have found the complete solution. The number of operations will be in the order of 50 to 100 for each type. Because of this, it cannot be just a "user-defined type", but must be crafted in a more general setting (DBMS manufacturer). Certainly, there is also a need for a more formal specification of their semantics. Related work is carried out in standardization groups for SQL-3 (SQL/MM) and Image Interchange (PIKS).

D1.2.2 Content Addressability (Content-based Search)

While it has been regarded as an important issue from the beginning [Chri85a], it has not been solved yet [Gros94a]. Pattern matching is useful, but does not help to answer all questions. On-the-fly analysis of media data with respect to a given query is nor possible with today's technology; image analysis and speech recognition are too expensive to be performed only as a part of query processing. Hence, description data (named "content-based attributes and relationships" in [Gros94a]) must be managed by the MMDBS and are used to answer queries. Keywords are the traditional concept for description data, but it does not suffice for the rich information content of multimedia data. New structures for description data (e.g. borrowed from knowledge representation) need to be examined. Their use must be offered to the user as a powerful comparison operator of the MADT, and their role in content-based queries must be understandable for the users, even if implementation details are hidden in the MADT.

D1.2.3 Revision of Extensibility Mechanisms

While new ADT's can be defined in extended relational and object-oriented systems almost equally well, the extensibility mechanisms are not suited for multimedia data. The run-time systems treats all those data objects rather uniformly, without taking their different characteristics into account. For them, text is the same as video, both are just objects with a certain set of methods defined on them. Instead, some sort of "handshaking" is required between the implementor of the methods and the run-time system so that the latter knows of the characteristics of the objects at hand. This is also known as "performance hints" in some systems. A related issue deals with the definition of access paths on the ADT's.

D1.2.4 Transactions

Even the semantics of updates on multimedia data is not a solved problem, not to speak of transactions, i.e. sequences of updates that must be performed in an atomic way. Neither locking nor logging techniques used in today's DBMS scale smoothly to the handling of MM objects. The granularity must be discussed first. It might well be that the proper granule is now at the sub-attribute level. Also, nested transactions might be the only choice to give the user (i.e. the programmer of the MM application) a clear understanding of what is going on in the system.

D1.2.5 Defining and Managing Relationships among Monomedia Data

(Capturing the "Multi" in Multimedia) Most data models offer a feature to define generic relationships among data objects. However, the semantics of these relationships is left to the application. In multimedia systems, very specific relationships are known to exist among the data objects. These are: Aggregation, synchronization, hyperlinking, substitution, derivation, to name just a few. It is not obvious whether it is sufficient to let the application handle them, or whether the DBMS should take their different semantics into account. Most of them are relevant in guiding navigational and descriptive access to the data.

D1.2.6 Architecture of an MMDBS

The issue has been raised by Masunaga already [Masu87a], but still has not been solved. The choice is between integrated systems managing standard alphanumeric data as well as multimedia data, and a set of media-specific, optimized, and interoperable systems. The issue of interoperability of media storage servers is also relevant in the context of network access (see below). There are some activities in the Berkom project to define a Multimedia Archive Teleservice.

D1.2.7 Network Access to an MMDBS

Resource allocation is a problem even in the centralized case, but it becomes more complicated in a distributed setting. Much work is carried out to develop multimedia communication services and protocols. They must be coordinated with the mechanisms developed in MMDBS. Global resource allocation has been investigated in the DASH project and in the HeiProjects.

D1.3 Research Agenda

The standardization groups dealing with SQL/MM and PIKS need input from the multimedia and database community. They have to exchange ideas and find a solution general enough to hold for some longer period of time. Formal specification techniques should be employed to guarantee a common understanding of the semantics of the operations. Since ADT's are designed, algebraic specification will be a candidate, but other techniques should be tried as well.

Media-object storage managers should be implemented and tested in isolation, before the extensibility mechanisms of advanced DBMS are revised. Those managers are necessarily interoperable, because they need the assistance of DBMS for alphanumeric data to be able to cover all the data management needs of the applications. They provide a testbed to experiment with the ADT operations, buffer management, transaction concepts, resource allocation, and content-based search.

Content-based search will be application-specific to a significant degree. However, generic mechanisms that support it should be investigated. One idea has been developed by the author, but has some known deficiencies. Other kinds of description data should be analysed and tested.

Finally, the issue of relationship modeling has to be addressed. The analysis of hypermedia models (HyTime, MHEG, HyperODA, Dexter) will yield a large collection of means to represent relationships. They must be classified and mapped to each other whenever possible. The result should be an encompassing concept of relationships for multimedia data. As a consequence, operations on these structures must be defined. While not needed in the context of HyTime and MHEG, they are required in a more general setting.

References

[Chri85a] Christodoulakis, S., "Multimedia Data Base Management: Applications and Problems - A Position Paper", in Proc. ACM-SIGMOD 1985 Int. Conf. on Management of Data (Austin, Texas, May 1985), ed. S. Navathe, ACM SIGMOD Record, vol. 14, no. 4, Dec. 1985, pp. 304-305.

[Gros94a] Grosky, W., "Multimedia Information Systems", IEEE MultiMedia, vol. 1, no. 1, Spring 1994, pp. 12-24.

[Klas90a] Klas, W., Neuhold, E., and Schrefl, M, "Using an Object-Oriented Approach to Model Multimedia Data", Computer Communications, special issue on Multimedia Systems, vol. 13, no. 4, May 1990, pp. 204-216.

[Klas92a] Klas, W., "Tailoring an Object-Oriented Database System to Integrate External Multimedia Devices", ICSI, Berkeley, CA, 1992.

[Masu87a] Masunaga, Y., "Multimedia Databases: A Formal Framework", in Proc. IEEE CS Office Automation Symposium (Gaithersburg, MD, April 1987), IEEE CS Press, Washington, pp. 36-45.

[Woel87a] Woelk, D., Luther, W., and Kim, W., "Multimedia Applications and Database Requirements", in Proc. IEEE CS Office Automation Symposium (Gaithersburg, MD, Apr. 1987), pp. 180-189.

[Woel87b] Woelk, D., and Kim, W., "Multimedia Information Management in an Object-Oriented Database System", in Proc. 13th Int. Conf. on VLDB (Brighton, England, Sept. 1987), eds. P.M. Stocker and W. Kent, Morgan Kaufmann Publishers, Los Altos, CA, 1987, pp. 319-329.

E Multimedia Networking and Communication

E1 Multimedia Networking and Communication

Derek McAuley, University of Cambridge, Great Britain

In the support of multimedia communications, there is often talk of the need for service guarantees. This paper presents an opinion on the current state of thinking in this line and presents the argument that currently there is insufficient separation of policy and mechanism to provide the required flexibility. A parallel with windowing systems is drawn.

E1.1 Introduction

Multimedia networking in a circuit environment is straight forward. Everything gets a guarantee of its maximum requirement; timeliness requirements are trivially dealt with. Why isn't multimedia communication a solved problem?

In the following, when I refer to an application, it is meant to imply a particular tool or set of tools, rather than a specific use of them. Hence a a pair of programs that source and sink video is a on of these generic applications, while a videophone is one particular use of the application with some added requirements like >5fps and at least 128x128 pixels.

A view could be taken that the reason for CCITT (now ITU-T) selecting ATM rather than STM for the delivery of B-ISDN was to enable increased profits by the use of statistical multiplexing while still alleging to provide ‘‘circuit like" guarantees. This strategy is already very successful in selling high cost bandwidth to several customers at the same time (e.g.~trans-oceanic), where silence suppression and TASI (time assigned speech interpolation -- i.e. compression) are used to provide the channel. However, as we have all experienced, 1) it doesn't always work as expected (i.e. it's not a guarantee) and 2) fax machines traffic completely break the system and must be dealt with explicitly -- fax being a traffic type that wasn't considered in the design.

Conversely statistical multiplexing has been traditionally used in packet switching networks where the assumption is that there is no where near enough resource (switching capacity and bandwidth) to satisfy the simultaneous peak demands of all customers. In this environment end-systems (transport layers up to applications) are expected to adapt their behavior to the available resources -- if they don't people stop using them. The current MBone tools (nv, vat, etc) must be the extreme case of such highly adaptable systems which attempt to provide multimedia functions with no explicit guarantees anywhere in sight. However, basic control theory shows that such adaption can be unstable without the application of some constraints.

These two views seem to be slogging it out. One suggests that guarantees can be provided as long as the application declares its requirements in advance with the benefit that the application is simplified as it doesn't have to engage in adaptive behavior -- the aim perhaps is to make an ATM network deal in ‘‘fuzzy" circuits. However statistical multiplexing leads to guarantees that are likewise statistical in nature.

The other suggests that the network shouldn't provide much more than best effort and applications can be made to adapt intelligently using the increased processing power available on every desktop. The latter is prone to instability and a reliance on the end-systems playing the same game.

Both camps have moved some way towards each other -- there is the suggestion of in-call renegotiation while in packet switching there are recent suggestions to provide hints to routers thus enabling them to provide a more stable service to designated packet streams.

At an absolute level, there is no such thing as a guarantee -- can you guarantee 64kbps in a circuit network, in the face of earthquakes, fire, famine, flood, pestilence... Where it is alleged that guarantees are required for particular uses of applications this is often a statement of the minimum service that this instance of the application is willing to accept.

E1.2 A Suggestion

I would suggest that while the network should be free to modify its service in an arbitrary manner according to its own policy, it should implement this in such as way that it provides applications with the information they require to adapt:

- networks should provide a service which is to guarantee not to modify the in-band part of communication more frequently than some agreed rate; hence appli-

cations know that any computation they perform to adapt their behavior is ‘‘cost-effective",
- networks should inform end-systems (applications) of modifications to their resources in a timely manner to enable them to take appropriate action, rather than having to rely on observational data (like TCP).

E1.3 Research to be Done

More work is required on adaptive coding schemes and environments in which it is possible to write adaptive applications -- this requires applications to know not just how much network capacity is available, but also bus bandwidth, processor cycles, memory, etc.

The network control mechanisms need to be more ‘open', in providing information to end-systems and applications on what is really going on -- using observation to deduce this state is inefficient and often the availability of information is delayed.

Further, when considering personal end-systems, mechanisms need to be in place to enable the user to overlay their policies on the applications given what the network is saying it can provide. In many ways this is similar to the use of window managers to control the computer desktop; given a fixed resource (pixels), as more applications are started and competition for the resource increases, the user makes policy decisions and enforces them via the window manager -- by the use of ‘‘vtwm" my desktop currently has nominally an over-allocation or pixels of about 10:1, but I have forced most of the windows to be un-mapped so I can see what I'm doing...

E1.4 Conclusions

I believe the graphics and windowing people have solved some difficult problems in allocating scarce resources and making it intuitive for programmers and users alike.

We should learn something from the techniques and protocols used in windowing systems for the allocation of pixels and colourmap entries; networking needs the concepts of expose, map, un-map, multiple colourmaps, iconize, etc...

E2 Multimedia Internetworking

Stephen Pink, Swedish Institute of Computer Science, Sweden

Multimedia Internetworking may demand changes to both the protocols and the user interface to the Internet. In the new Internet, real time applications may be served explicitly by protocols that reserve resources and that forward packets using switching as well as traditional Internet routing. The new concept of "flow" is examined and various philosophies of resource reservation are discussed.

E2.1 Introduction

The goal of working on multimedia networking, in my opinion, is to integrate real time voice and video applications into the distributed system platforms that we are all used to. This means being able to have audio and video teleconferences, listen to music and watch TV, all at my workstation. Since I envision all of this integration happening in the future (it is going on now, of course) I have to imagine that my workstation is going to have a faster CPU and lots more memory and probably more disk. I also imagine that my workstation is going to be attached to a high speed network (> 100 Mb/s). The question for this position paper is whether and how the network will have to change in order to do this integration.

An important constraint in answering this question is that the network will have to work for traditional applications just as well, after the integration with multimedia takes place. Right now, the network seems to work just fine. By the network I mean the TCP/IP protocols that I use to communicate on the Internet. So, the question becomes, how must the Internet change to support integrated services.

E2.2 A Changing Internet

The Internet is changing in at least two ways. First, a lot more people are becoming connected, and second, these people will want to use real time multimedia. (These two phenomena are not independent.) By lots of people, I mean that almost everyone in the US will be connected in the next five to ten years. By real time multimedia, I mean that on many Internet links, people will be want things like video on demand.

E2.3 Changing the Internet Interface Model

Thus, there will be pressure on the Internet to change its service interface so that at least two kinds of service is provided to applications. One for real time applications (that are delay-constrained), and one for the rest, although it is suggested that there will be more than one kind of service for real time applications (cf., Clark, Shenker and Zhang in on predicted and guaranteed service [1].) Some people think that it is possible to keep the same interface as now (i.e., single-service) and just change the queueing disciplines in the gateways to accommodate the new traffic.

It would be nice if the latter were true. It seems easier to change the internals of the network and keep the interface the same than it is to ask applications to be rewritten. But it may not be all that hard to do this if the change to the interface comes soon and is restricted to the applications that are delay constrained, because there are not all that many multimedia applications around right now. In that case, traditional network applications that use the implicit FCFS service in the network could remain unchanged.

Among those who feel that the user interface to the network must change, there is a lot of talk about the user “passing a flowspec” to the network that characterizes the resources needed by the application. This flowspec could then be used as the basis of negotiation between host and routers to establish a flow that has an associated quality-of-service. This quality-of-service would have some state associated with it in the network. This view has become so prevalent, that the leading candidate for the next generation IP, SIPP, has a field in the header called the flow id that informs hosts and routers that a SIPP datagram is part of a flow that could be established in the network.

E2.4 What is a Flow?

Just what sort of creature is this flow? Is it a connection in disguise?

The Internet culture does not like the word “connection.” Connections are things that phone companies use to establish communication, and the Internet feels that not having connections is one of the main reasons that Internet has been able to scale to the numbers of users that it has been able to support so far. I work for a research institute that has a national telecom company and telecom provider as its major sponsors. When I came to Sweden, and applauded the growth of the Internet there, the foeSuks from the phone company told me that the Internet was just a toy network because it didn’t have the 10,000 guys who were awake 24 hours a day to make sure its “switches” always stayed up. My response was always: the phone network needs these 10,000 guys because communication is connection-oriented in that network. The communication path is fixed once and for all at connection establishment time, so the switches better stay up always. The Internet, by contrast, is connection-less. If a router goes down, the datagrams can take another path. So the Internet has its fault tolerance built into its design.

E2.5 Soft-state

But real time services might demand that packets always take the same path, at least take paths that have the same delay. The new Internet architects, then, have come up with something that is between a connection-oriented or virtual circuit network and a pure datagram network. These are flows. A flow is something that has “soft-state” associated with it in the network. By soft-state it is meant that although there is no connection establishment end-to-end, there is a path established temporarily upon which datagrams are switched. This path can time out and be refreshed by new path messages that work their way through the network. And, although, path messages will start at the sender and work their way to the receiver, reservations will be made by the receiver and travel “upew Numm” to the sender.

E2.6 Connections and Reservations

This decoupling of path establishment and resource reservation is at the heart of this new architecture (RSVP [2] with multicast IP or SIPP [3]). Protocols such as ST-2 [4] and ATM require path establishment (connection establishment, in the terms of these latter protocols) and resource reservation to happen at the same time.

This makes network scheduling very inflexible. For suppose that some people in this room, working on different continents, plan a week ahead to have an important audio/video teleconference. Much preparation is made, and on the day of the conference, at the planned hour, each user dials up the other on their ST-2 or ATM network node.

Now the resource reservations schemes proposed for these networks usually have a component of call admission, and on this particular day, the circuits are busy and the teleconference has to be postponed. This seems a shame, since the users knew a week ahead of time that the conference was going to take place, but there was no way in these architectures to get that early reservation and a possible accompanying price break from the provider who could have better scheduled the network with their advance reservations. No one these days expects to get a seat on a transatlantic flight by just showing up at the airport. Rather we make "APEX" reservations at least a week ahead and get a price break for them. In the network, with advance reservations, we might still pass a flowspec at connection establishment time, but that flowspec will stand to our advance reservation like a boarding pass stands to the original phone reservation we made a week earlier. The problem is that putting advance reservations in the network, although potentially offering a more efficient service, seems to increase the state the network must keep into a third dimension: time. Thus, since advance resource reservation on the connection-oriented model implies so much complexity, it might be better to abandon that model in favor of one that will scale better, yet offer the same service.

E2.7 The Mobbing Model

At present, in the Internet, there is a program called "sd" (session directory.) This program, which runs on hosts in the Mbone, is a place where future IP multicast teleconferences can be advertized. This handy little program allows a user a point-and-click interface to audio, video and whiteboard sessions, via the "vat", "nv" and "wb" programs, that can be used with multicast IP on the Mbone. (The Mbone is an IP multicast routing superstructure that uses the current Internet for tunneling of multicast IP datagrams.) Knowledge of future events is propagated between "sd" processes on hosts via IP multicast. Since "sd "contains future Mbone events and their duration and the bandwidth in the Mbone tends to be rather scarce at present, "sd" provides a good clue as to whether a future teleconference should be scheduled. One can imagine an enhanced "sd" program that provides a statistical view of the traffic in the network and functions as a kind of "call admission" mechanism for the Mbone. If this were to work, then the network itself would not need to keep nearly as much state as on the connection-oriented model, since knowledge of present and future use of the network would only need to be on end users' workstations.

E2.8 Routing and Switching

Our original question was about the nature of flowL in the new Internet architecture. FlowL seemed to be a third thing, different from virtual circuits and datagrams. Actually, the flow architecture is a hybrid of the two. In each SIPP packet, there is the source and destination address (which is the defining aspect of the datagram model) and an optional and shorter flow id, which is what one often finds in a virtual circuit network. Very roughly, in RSVP, if a valid flow id is not present, then the packet is routed as in the current Internet. If the flow id is valid such that there is state in the routers associated with the flow, including next-hop as well as queueing priority information, then the packet is “switched” on the flow id. This state information is created in the routers (routers can now be switches) by a path message which itself is routed through the network. Of course, the first path message could find a non-optimal route. But path messages are sent more-or-less freq the partly as state times out, so non-optimal paths can be corrected. The ratio between the number of path messages and data messages sent is the overhead of this scheme.

Thus, the new architecture is a hybrid of routing and switching. The default method of forwarding packets is routing, since every datagram will contain source and destination addresses. For communication that needs quality of service, flows can be established and packets will be switched on the flow id. But the source and destination addresses will be present to guarantee robust network control. This last is a point worth emphasizing.

E2.9 Datagrams and Virtual Circuits

During our implementation of the ST-2 protocol [5], we posed an experiment to eliminate the “hello” messages from the protocol. We thought that adjacent ST-2 routers and hosts pinging each other at regular intervals was a potential waste of bandwidth and processing time. We found, however, that “hello” messages were essential. ST-2 has only a virtual circuit id (i.e., “hop id”, like a flow id) in its data packets. (Source and destination addresses are only contained in control messages.)

Due to problems with the network, sometimes receiving hosts and gateways would crash and then become unable to boot. This was because sending hosts would continue to send data packets, which contained only virtual circuit id’s and no source addresses, at receiving hosts at high rates. Since the crash eliminated the mapping between the virtual circuit id’s and the source addresses, there was no way for the receiver to turn off the sender. If we had implemented the “hello” messages, then the sender’s “hello” message timer would go off after not receiving any acknowledgment from the receiver, and the sender would know that a crash had occurred and stop sending. If the ST-2 data packets had both flow id and source and destination address in them (as in SIPP), the receiver would be able to stop the sender from deluging it with unwanted packets without the need for a “hello” message mechanism. The trade-off, of course, is header size: it would not be a good idea to add another 8 or 16 bytes of address to a small cell. But for a variable length packet, having these addresses seems well worth the space because of added robustness. And if forwarding on flows is done on the flow id, then the large addresses that accompany each packet do not figure in the forwarding costs.

E2.10 Conclusion

In this position paper, I have brought up some of the issues involved in changing the Internet to provide real time service to multimedia applications. I have discussed reservation mechanisms as well as changes to the architecture that involve a new concept of flow in the Internet. Introducing flows into the architecture, it was suggested, also introduces some mechanisms that are common in virtual circuit networks as well as new mechanisms such as soft-state. Finally, I have compared some traditional connection-oriented protocols such as ATM and ST-2 with a new set of Internet protocols, RSVP and SIPP, designed to provide soft-state in the network without requiring traditional connections.

References

[1] D. Clark, S. Shenker and L. Zhang. Supporting Real-Time Applications in an Integrated Services Packet Network: Architecture and Mechanism. In *Proceedings of ACM SIGCOMM*, pp. Element4-27, Aug. 1992.
[2] L. Zhang, S. Deering, D. Estrin, S. Shenker, and Daniel Zappala, RSVP: A New Resource ReSerVation Protocol, *IEEE Network*, vol. 7, no. 5, September 1993.
[3] S. Deering, Simple Internet Protocol, *IEEE Network*, vol. 7, no. 3, May 1993.
[4] C. Topolcic. Experimental Internet StrNumm Protocol: Version 2 (ST-II). Internet RFC 1190, October 1990.
[5] C. Partridge and S. Pink. An Implementation of the Revised Internet Stream Protocol (ST-2). *Internetworking: Research and Experience*, vol. 3, no.1, March 1992.

E3 Future Integrated Storage/Communication Architectures

Radu Popescu-Zeletin, Stefan Covaci, GMD-FOKUS, Germany

Advances in optical networking technology provide the communication network with a new quality, i.e. storage of huge amounts of data. Additionally, these improvements widened even more the gap between the traditional data processing/ communications and telecommunications worlds, in that a number of bottlenecks are gaining in significance and some new ones appear, affecting overall performance. Distributed processing, mobility, multimedia-based applications and the need for global networking call for a harmonization between processing and communication. The paper presents two scenarios of a target data processing/communication architecture that alleviate the main identified bottlenecks and pave the way towards a global distributed system supporting multimedia-based applications. The proposed processing/communication paradigm is based on the shared-memory capabilities offered by the optical network. A new structure referred to as the network-memory is introduced and used as building block in the system architecture. There is a need to reconsider the roles played by the "network" in this new environment, as well as the notion of "end-system". Possible solutions for the network-memory and for the end-system architecture are given, and related issues are analyzed.

E3.1 Introduction

With the dramatic improvements in optical networking technology a series of performance bottlenecks in communication are becoming increasingly important, and some new ones can be identified. They are due merely to "traditional" mechanisms devised for data communication at times when the network was limiting overall performance.

Optical technology offers huge transmission capacity, and a low error probability due to reduced crosstalk and intersymbol interference, as well as good noise immunity. These of course will have a considerable impact on the kind of mechanisms used for transmission media access and for error control.

The message segmenting technique was merely introduced for sharing a limited resource, i.e. the network bandwidth, and for fast error detection/recovery in an error-prone media. Support of several priorities was also a reason behind slotted traffic. The new fiber optic environment gains a totally new dimension - practically infinite bandwidth available, at very low error probability. The unique capacity property of the fiber provides the network with a new quality, i.e. the network becomes an optical memory.

A first bottleneck is the optical/electronic interface. There is an almost four orders of magnitude discrepancy between peak electronic processing speed (a few Gbps) and fiber capacity (tens of Tbps). This motivates developments towards optical processing but also developments of new multiple-user access mechanisms (e.g. WDMA) [1], [2] and switching/multiplexing systems. The multidimensional nature of optical systems (frequency, time, space), combined with their bandwidth capabilities, offer the possibility of building compact switching systems supporting considerably more channels than their electronic peers.

A second bottleneck one may observe is the switching technology adopted. The new ATM networks are supposed to integrate data communication and continuous media in one network. The transmission technology between ATM switches is based on SDH, providing synchronous channels with guaranteed quality of service. The ATM network provides statistical multiplexing in the network, delegating the end-to-end provision of the required quality of service to the end systems. Complicated procedures have to be devised to ensure the same qualities of service (guaranteed throughputs and jitters) as those provided by the transmission network.

The third bottleneck is the attachment architecture of the end-system to the network. In fact the communication subsystems or the host-network adapters are gateways between a workstation´s internal network (bus, channel, backplane) and an external communication system. The behavior of the internal network and of the external one are in general not compatible, from the access mechanism to the protocol data unit. Even with hardware implementations and parallelization of protocol machines [3], [4], a major loss can be expected due to a missing coherent architecture of the network and of the end-system. Protocol processing optimization leads usually to minor performance gains.

The forth bottleneck appears when one tries to integrate the two existing worlds of data and continuous media (audio, video) communication. Data communication is characterized by its requirement for high reliability, which is usually reached by asynchronous communication using a layered architecture and hand-shaking protocols. This protocol stack processing which deals with endless protocol data units conversions, with CRC calculations, and with acknowledgments, makes the whole communication software to the major bottleneck.

A rethink of the mechanisms needed for error protection is necessary, not only because of the better noise figure of the optical fiber, but also because of the impact of the communication latency factor due to light propagation, which at very high transmission speeds and long distances will essentially require for Forward Error Control (FEC) [5].

Continuous media applications on the contrary, require low reliability but isochronity of the unidirectional transmission. The general question here is which is the right paradigm when aiming to integration, the data communication one or the continuous media one, or in other words which paradigm has to be kept.

A fifth bottleneck is the storage capacity and the access time to the secondary storage, in response to the continuous media requirements. The data processing architectures are based on shared memory, and storage capacity and the I/O bandwidths are limiting factors for processing performance.

The sixth bottleneck is caused, similarly to the third one, by a noncoherent integration of the communication software with the end-system´s operating system. Protocol stacks for continuous media require guaranteed resources, and thus resource reservation. Based on the adopted solution protocols are implemented in the user space or kernel, and usually concur with other processes for resources. General purpose operating systems do not provide the necessary mechanisms to ensure guaranteed resource allocation for continuous media communication.

More significant than protocols interpretation is the overhead of the memory system caused by data movements, and the overhead introduced by the operating system when switching context and copying data.

These bottlenecks can be treated individually but with great effort, and cannot be fully removed.

Historically, end-systems and networks developed independently and there is little synergy between their architectures. Additionally, the technological developments in the data processing/communication and in the telecommunication arenas were not correlated (due to little common interests) and lead to quite different solutions.

The situation has changed and the current users´ needs call for a harmonization (marriage) between the two worlds. Driving forces towards such a harmonization are among others, the need for a global network supporting multimedia-based communications, integration of data/audio/video, distributed processing, mobility, and collaborative environments. Increase of processing and communication performance is needed.

As mentioned before, bottlenecks can be identified and treated individually but this will bring little synergy between the two worlds. It is perhaps worth to rethink the whole environment and devise a target solution common to both processing and communication. In fact in the new evolving processing/communication environment the network is no more confined to the LANs and PNO´s realms but includes also the end-systems with their communication capabilities. With the advent of metacomputing, communication becomes part of the processing, and the communication paradigm changes.

E3.2 System Architecture

Analyzing the identified bottlenecks and the characteristics of the today´s network technologies one may question which is an ideal target data processing / communication architecture for the future. The characteristics of such an integrated system should melt together the end-system and the transmission / networking technology. The new network technologies exhibit some fundamental characteristics:

- very high data volume underway (high throughput)
- low signal attenuation combined with recent amplification technologies
- very low error probability
- switching techniques at lowest level (e.g. WDM)

From the first two characteristics the network can be viewed as a shared memory with considerably more storage capacity (e.g. recirculating data on a perfect shuffle) than the capacity available in the end-systems. Additionally, it provides much higher bandwidth than the internal busses and backplanes.

We attempt to reformulate the processing / communication paradigm and sketch some scenarios and possible solutions. The aim is to consider the network and the end-systems as a distributed system based on shared memory.

The network can be viewed as a bus on which the processing units of the end-systems are attached. The additional property offered by this bus is its huge storage potential. The network integrates the functionalities of memory shared by, and communication between the distributed processing units.

The several processing units will use a single data structure for I/O operations which will correspond to the one in the network (internal data structure of the memory).

The *write time* into this memory is the media access latency, and should be possibly brought to zero taking into account the "unlimited" available capacity and the multidimensional nature of the system.

The *read time* is essentially determined by the distance (duration of light propagation) between the read tap and the data packet underway in the network and, in the worst case is equal to the round trip propagation time.

In order to satisfy the QOS requirements of different types of applications (data processing, continuous media) on one hand, and for simplicity and controllability on the other hand, we are targeting in the design of the system towards a single service, satisfying the highest requested QOS. Two scenarios can be envisaged:

(a) one shared memory to which all processing blocks are attached. This is referred to as the no-cache model, and
(b) a multitude of interconnected shared memories. In this scenario a cluster of processing blocks access a local shared-memory (the cache) and form a first level of the storage/communication hierarchy. These clusters interact with other ones via the backbone memory - the next level of the storage/communi-

cation hierarchy.

Essentially, from the service point of view (storage), there is no difference between the two. Functionally, there is a single memory shared by all processing units.

Traditionally, the notion of an end-system is defined by the locality of its components with respect to the network. In the proposed paradigm this view of an end-system essentially disappears since its components can be geographically distributed, and depending on the user application reconfigured every time. For the by the application grouped processing units, the network can be considered as a common bus offering shared memory functionality. This is not new, smaller steps towards this redefinition of the notion of end-system may be observed in the developments of the notion of server (e.g. file server, storage server) in a network where several remote situated services are integrated in the end-system [6].

When long geographical distances are to be covered, the latency due to light propagation becomes the limitative factor for satisfying required QoSs, and little can be done to overcome this limitation [5], [7]. In this case processing should be organized in a way that considers the obtainable memory access times, and eventually requires for scenario b). This is not the case with a local shared-memory where several techniques can be envisaged to obtain the desired memory access time independent of the needed storage capacity (see section 4).

We may say that it is not the capacity but the distance spanned by this memory that limits the access time (read component). The proposed system architecture alleviates the following bottlenecks:

- bottleneck 2, by implementing a single communication service satisfying the highest QOS required (e.g. no need for several AALs when using ATM as multiplexing technique)
- bottleneck 3, since there is only one communication system (mechanisms) between the processing units. The so called Host- Network Interface (HNI) does no more play the role of a gateway between two networks, but will be an I/O device to the shared-memory, implementing the same service for all units.
- bottleneck 4, since it is able to provide a reliable isochronous communication service. Scenario b) will enable implementation of latency-hiding techniques (caching techniques and other anticipation mechanisms) in order to keep the level of QOS even when long distances are involved (see next sections).
- bottleneck 5, since this new shared memory offers considerable more storage capacity and higher bandwidth for the new class of I/O operations. More than that, the I/O bandwidth can be controlled and pushed close to the available optical transmission one, by use of several technological and concurency-related solutions.
- bottleneck 6, since it enables the design of a new flexible OS that is releaved from the overhead related to pure communication (data copying, context switching,...) and of the task of arbitrating accesses to limited resources (I/O bandwidth, bus bandwidth, memory). The system architecture enables a more DMA-

oriented operation, where the OS will have to be adapted to this new session-dependent, distributed end-system environment.

The idea to use the distributed memory system for interprocess communication as an alternative to the message-based systems has been mentioned in [7] (again separation between communication and processing), but in our proposed architecture a higher degree of merger between the distributed memory blocks and the communication channel that links them is achieved, in that there is a single structure that is used as memory *and* as communication system, and this is the optical network-memory.

Based on the double functionality offered by this structure, other synergetic factors are becoming possible. Additional issues related to the alleviation of bottlenecks 1, 2, and 5 are treated in more detail in the next sections.

E3.3 Possible Solutions for the Network-Memory

The role played by the optical network in this distributed architecture is twofold: shared-memory and communication system. The main characteristics of this memory are the storage capacity and the access time.

E3.3.1 Storage Capacity

The storage capacity is related to the light-propagation time. At a data modulation rate of R bps, there is need of 300,000 km of light propagation in order to store R bits by using only one wavelength. The longer the propagation latency, the higher is the obtained storage capacity. We are faced here with the problem of *increasing the propagation latency,* as opposed to the need to reduce this latency when optimizing the communication functionality of the network.

The problem of obtaining long light-propagations must be considering the need of a small formfactor when implementing a local memory (scenario b). The multidimensional nature of light is here exploited in that a needed propagation length of S can be reduced to S/n by using n wavelengths. Related technological influencing factors are channel separation, range and speed of tunability of optical sources and receivers.

An alternative approach can use independent wavelength channels shaped as parallel rings (memory pages), whereby in this case the mechanism to access a specific channel has to be provided by an optical element acting as interface with the external user. Such a structure will be shaped in direct correlation with the organization of processing.

Light can propagate in a fiber or in a low-loss media that favorizes the increase of the propagation length. Multiple reflections in a glass-cube could be a possible solution, and it is the crosstalk and the optical-loss components (like amplitude and chromatic dispersion) that will merely influence the formfactor.

The distance δ is merely imposed by the crosstalk between adjacent propagation planes, while α is a parameter limited by technological factors.

More efficient methods based on pure reflections can be imagined for the change of the propagation plane. Regeneration/amplification blocks as well as other elements can equip such a device in order to compensate the dispersion and power losses. Read taps can be placed along the propagation path in order to obtain faster access to the data underway.

For the backbone memory, the storage capacity is a direct consequence of the fiber length.

E3.3.2 Access Time

This characteristic is influencing not only processing but also the obtainable QOS of the data communication service implemented by this memory. The smaller the access time is, the better is the obtained overall performance. Memory access time has a write and read component.

The *write time* into this memory is the media access latency, and should be possibly brought to zero taking into account the "unlimited" available capacity and the multidimensional nature of the system. Several solutions can be devised and analyzed to this purpose (multi-user access protocols, e.g. WDMA, topology issues, etc.), but this out of the scope of this paper.

The *read time* is essentially limited by the duration of light propagation. When referring to the local memory, one can tap the light-propagation in such a way that the desired read time is obtained for the local processing needs (see Fig. 3 and Fig. 4). There is here an analogy with a fixed-heads disk drive where the access time is a function of the number and position of the read/write heads.

When remote processing is involved, the read time is subject to backbone-topology optimization. We are here facing the problem of minimizing and hiding the light-propagation latency [7].

Memory functionality like the buffering for a predetermined time can be obtained by devising data rotation mechanisms, which control the rotation time, i.e. the number of round trips of a data unit in the memory.

E3.3.3 Network-Related Issues

Considering the last two characteristics of the network technology and the aspects discussed before, several issues have to be analyzed:

- services (asynchronous, isochronous),
- types of traffic (connectionless - CL, connection-oriented - CO)
- media access mechanisms (multi-user, concurrence),
- multiplexing / switching, topology (shared media, bus-, switch-based),

- protection against errors
- information transmission (fixed-size cells, variable
- length packets)
- addressing

As already mentioned, the target network should offer a single service at the highest QOS required. Asynchronous and isochronous services differ with respect to *the value* of the jitter of the time period between successive data transmissions. Isochronous services are asynchronous services for which the jitter does not exceed a by the user imposed upper limit. In our case this limit is the minimum time period between two successive memory-write operations requested by a single user. For our scenario the network should provide the service that satisfies the most demanding user (e.g. human perception of isochronism when transmitting an audio/video document).

The grade of service is strongly influenced by the media access latency and by the light propagation latency. The first factor should not cause severe problems since there is enough storage capacity available. The second one will influence the "real-time" aspects of an application and the granularity of the "distributed processing", and consequently will determine the classes of supported applications and the overall organization of processing.

The aspect of CL/CO is not that important at this stage of the analysis. Nevertheless, when implementing a local cache, the network should provide its users with a CL service since in this environment a very small access time to the memory can be obtained, and because CO requires signalling (extra complexity, adds the establishment/clearing time).

The network must provide concurrent access to multi-users in order to behave as a shared, multi-ported memory. Concurrence will be supported by MAC mechanisms and by multiplexing techniques. Also multicast, and group addressing have to be supported and will be used to shape the internal organization of the memory and to provide ownership and access protection mechanisms.

Protection against errors will be achieved using a single mechanism (one service). The choice of this mechanism has to consider the propagation latency factor and thus may achieve several degrees of data integrity. For the long propagation network, reactive mechanisms (feed-back loops) are in general not applicable, and FEC techniques should be implemented. For the local cache some reactive mechanisms can be additionally implemented when higher degrees of protection is needed.

The media can be slotted (fixed length cells) or can transport variable length packets. This has an impact on the media access latency and on the support of traffic priorities. Since a single priority traffic is supported there is no need of *small* cells. Nevertheless, slotted media permits implementation of MAC disciplines that can control access latency and fairness. On the other hand these may no longer be so important when enough bandwidth is available. A more important influencing factor is the traffic characteristics (burstiness, needed capacity, number of users) which is applications dependent. For instance, statistical gain is no more a main issue when considering the today´s metacomputing applications involving highly bursty transmissions of high data volumes among few users. In this case a variable packet length transmission mode is preferable. In our scenario we estimate a great number of users with different capacity requirements, so that statistical gain may be needed at a certain development stage. This is why a slotted media is preferable, but perhaps with a larger payload than in ATM cells (e.g. capable to transport a HDTV video frame). The argument on using a variable length packet transfer mode needs further investigation.

The network aiming to be global must implement, for several considerations, a synchronous digital hierarchy.

Addressing schemes will have to accommodate a very large number of users, and a hierarchical scheme like E.164 is an appropriate approach.

E3.4 End-System Architecture

In order to integrate the network-memory in the distributed processor we need a single data structure for all attached processing and I/O units. This will be correlated with the format used in the network. No more format conversions will be needed. Present systems use several data structures. On the other hand, different QoSs required by the network users lead to different data structures in the communication part, and thus caused performance bottlenecks due to necessary format conversions.

For scenario a) the end-system will consist only of processing units and classic I/O devices (displays, printers, video cameras, microphones, non-volatil storage devices, etc.) grouped together in order to execute a task.

For scenario b) the end-systems will have also local-caches. These will also be optical network-memories. The main characteristics of the local-cache are storage capacity, access time for local processing requirements, and reduced physical dimensions (small formfactor). Technical solutions for obtaining the desired values for the first two can be envisaged, and to some extent for the third one.

This cache will be connected to the network-memory outside the end-system in order to interwork with remote processing units. If we imagine the cache implemented in the form of an optical ring to which all local processing units are attached, then this ring is functionally closed when local processing takes place, and opens itself and ties its ends to the outer ring when remote processing is required.

New operating systems have to be devised in order to support this session-oriented distributed processing and they need to be flexible enough to configure and control the new type of end-systems enabled by the architecture.

E3.5 Conclusions

We made an attempt to reformulate the data processing / communication paradigm by proposing an new architecture for a global distributed processing system based on shared-memory. The shared-memory is implemented by an optical network, and thus enables a new structure referred to as network-memory, that offers simultaneously memory and communication functionality. The new system architecture alleviates a series of the identified bottlenecks due to a high degree of harmonization between data processing/data communication/telecommunication, and favorizes a reconsideration/redefinition of the notion of end-system. Optical technology gives us hope that network-memory structures with very high storage capacities and short access times are feasible. A series of important related issues like Operating Systems, MAC mechanisms, multiplexing/switching, topology, propagation-latency hiding, and organization of processing need further attention and will be part of our future work.

References

[1] B. Mukherjee, "WDM-Based Local Lightwave Networks Part I: Single-Hop Systems", IEEE Network, vol. 6, no. 3, pp. 12-27, May 1992

[2] B. Mukherjee, "WDM-Based Local Lightwave Networks Part II: Multihop Systems", IEEE Network, pp. 20-31, July 1992

[3] B. S. Davie, "An ATM Network Interface for High-Speed Experimentation", Procedings of IEEE Workshop on the Architecture and Implementation of High Performance Communication Subsystems, Tucson, Arizona, February 17-19, 1992

[4] C. Brendan, S. Traw, J. M. Smith, "Implmentation and Perforance of an ATM Host Interface for Workstations", Procedings of IEEE Workshop on the Architecture and Implementation of High Performance Communication Subsystems, Tucson, Arizona, February 17-19, 1992

[5] L. Kleinrock, “The Latency/Bandwidth Tradeoff in Gigabit Networks”, IEEE Communications Magazine, pp. 36-40, April 1992

[6] David J. Farber, "Distributed Computer Systems: Backwards towards the future", Late paper for the Proceedings of ISADS, March 30-April 1,1993, Kawasaki, Japan

[7] J. D. Touch, D. J. Farber, "Reducing Latency in Communication", IEEE Communications Magazine, Vol. 31, No. 2, pp.8-9, February 1993

F Multimedia Documents and Mailing

F1 Multimedia Documents and “Live Data”

Simon Gibbs, Université de Genève, Switzerland

Abstract: The essence of multimedia lies in combining and composing elements from diverse media. To understand multimedia then, we must first understand composition. Multimedia documents provide a valuable context in this regard since many forms of composition are supported by the various document preparation and authoring tools. This position paper summarizes the predominant composition mechanisms now in use and identifies a new mechanism suitable for “live data.”

F1.1 Composition Mechanisms

Several generic composition mechanisms have been identified in the literature (by “generic” we mean mechanisms that apply to a variety of media) and are now found in many multimedia authoring tools. These generic composition mechanisms are:

spatial composition One of the most common forms of composition is the positioning of media elements in 2D or 3D space. Examples include document layout and graphics modelling. Spatial composition often results in a hierarchical structure – the leaves refer to media elements and the intermediate nodes specify relative positioning and rendering information.

temporal composition The composition of time-based media involves the positioning of elements along a temporal axis. Positioning may be absolute, as when explicit “start times” are given, or relative, as when one element starts after another finishes.

semantic composition Explicit links between related material is an example of semantic composition. The end points of links may be media elements or composites of some sort.

procedural composition With scripting languages, associations between media objects are expressed procedurally (or using a mix of procedural and declarative statements). For instance, parts of an image may be made sensitive to user input. When the region is selected a procedure is activated which might then play sound and animation sequences.

The four composition mechanisms are summarized in Table 1. In each case an associated *composition metaphor* is given. For example, spatial composition fits a document-oriented view of multimedia while temporal composition has the “movie” metaphor.

Table 1. Composition mechanisms and metaphors.

mechanism/ metaphor	*elements*	*composite*	*representation*
spatial/document	document elements (text, image, graphics objects)	multimedia document	tree
temporal/movie	tracks (audio, video, music, animation objects)	multi-track aggregate	linear
semantic/web	nodes (media objects in general)	hypermedia web	graph
procedure/script	“cast members” (media objects in general)	multimedia script	various (linear, graph)

Although identified above as four distinct mechanisms, in practice they are often mixed. Many examples can be given:

- spatial plus temporal composition: document systems supporting video components (e.g., a Microsoft Word document with an embedded QuickTime movie)
- spatial plus semantic composition: document systems supporting cross references (e.g., FrameMaker)
- temporal plus procedural composition: time-line based authoring systems which support scripts on cast members (e.g., MacroMind Director)
- semantic plus spatial composition: any hypermedia system
- semantic plus procedural composition: hypermedia systems with “virtual” nodes (e.g., World Wide Web HTML pages generated by shell scripts)

One can expect “rich” document models and authoring environments to support most, if not all, of the above mechanisms. In particular, current standards activities

(MHEG, HyTime, HyperODA), and commercial activities (e.g., ScriptX) combine several composition mechanisms.

F1.2 Live Data

Despite the generality of the spatial, temporal, semantic and procedural composition mechanisms, it is possible to find examples of multimedia which rely on yet other forms of composition. For instance, consider a live performance where a group of musicians simultaneously play their instruments (admittedly this involves composition in a single medium – audio – but the example can easily be extended to include other media). Now consider a digital representation of the performance, a “performance object” if you will. In “authoring” such an object the four composition mechanisms listed above could all come into play:

- spatial composition – the musicians are located in a 3D space, their positions and the spatial characteristics of the environment affect how the performance is perceived
- temporal composition – there are timing dependencies between performance events, the musicians share a common tempo and try to remain synchronized
- semantic composition – one could have links from the “performance object” to other multimedia objects, such as representations of the composer, the musicians, and the instruments
- procedural composition – the musicians may be following a score (a script), they may be responding to cues from a conductor

Although each of the composition mechanisms plays a role in constructing the “performance object” none seems to capture the essence of performances – the fact that they are *live*. Instead what seems more appropriate is a representation based on “producer objects” (the musicians and their instruments) and a set of *live data* streams (the audio values being emitted by the producer objects) which are, ultimately, combined (composed) in the ear of the listener. This representation can be seen as structuring the performance into functional units and can easily be extended to situations where, in addition to producers, there are “transformers” (such as audio mixers) and “consumers” (such as recording devices).

F1.3 Live Data Sources

Spatial, temporal, semantic and procedural composition offer many possibilities for combining media but share one restriction: they were conceived in, and are usually realized in, a context of stored, rather than live, data.

What is the difference between stored and live data? One way to view the distinction is to consider servers and clients, or, to slightly change terminology, data sources (e.g., file systems, data bases, video servers) and data sinks (e.g., presentation systems, browsers, viewers). With a stored data source the sink is “in control” – it has the choice of selecting what data to receive (subject to authorization constraints etc.) and when to receive it. With live data the situation is reversed, now the source is “in control” and data sinks have little choice in what is sent their way. (A simple example of this would be a “server” with *no* client interface, one that simply sends data out onto the network.)

With live data sources, applications loose control, so live data applications may not sound particularly flexible or useful. However there are many situations when live data sources may be more efficient or more timely than stored data sources. Examples of live data sources include:

- news feeds (e.g., Reuters)
- Internet broadcasts and multicasts (e.g., MBONE)
- digital TV satellite broadcasts
- digital TV cable broadcasts
- analog broadcasts of close-captioned TV
- digital cable radio broadcasts
- digital cellular radio broadcasts
- local-area IR broadcasts
- sensor streams (telemetry data)

F1.4 Composition of Live Data

Applications are not completely powerless regarding live data. Certainly they can choose whether to “tune in” or ignore a live data source. Of more interest is the operation of *filtering*. An application may only be interested in a small portion of a live data stream, or may want to transform the data in some way before presentation – the answer is to filter, i.e., process, the stream.

It seems then, that in describing live data, one has three basic constructs: sources, sinks, and filters. Sources produce live data, filters transform or process live data, and sinks consume live data. Suppose we call these *components*. Components are the basis of the additional composition mechanism shown in Table 1.

Table 2. Component-based composition.

mechanism/ metaphor	*elements*	*composite*	*representation*
component-based/ circuit	components (sources/producers, filters/transformers, sinks/consumers)	component network	graph

It should be possible to construct multimedia documents for the selecting and viewing of live data by composing (connecting) filter and sink components. When such documents are then connected to live data sources they become active and proceed to process and display the incoming data streams. For example, consider a document that displays *current* stock prices and exchange rates. Using stored data, the document would continually query some database. Using live data, the document connects to an appropriate source (e.g., a financial data feed) and then extracts the relevant information. Component-based composition can be combined with the other forms of composition, also multimedia documents should be able to combine live data and stored data. For instance one could have a live data window placed within a stored data frame.

Currently, component-based composition is not supported, at least to the author's knowledge, by multimedia authoring systems. What is first needed are standards for constructing and naming live data sources.

In terms of interactivity and presentation, component-based composition offers little that is new. Perhaps the importance of this technique is that it allows us to create documents that can be "patched in" to new network services. These new services, incorporating broadcasts and multicasts from live data sources, are emerging as bandwidth increases and protocols evolve.

F2 Multimedia Archives as Multimedia Mail and Document Servers

Erich J. Neuhold, Thomas C. Rakow, Heiko Thimm, GMD-IPSI, Germany

In this position statement we set the framework for a distributed multimedia archiving system that will be needed to support multimedia hyperlinked documents and synchronous and asynchronous cooperation via high speed networks.

F2.1 Introduction

In the (short) history of multimedia we have lived with essentially two separate worlds. One was concerned with the digital representation and exchange of all kinds of information, e.g. texts, graphics, images sound/speech, video. But if at all digitized these representations were kept separately, even on different media and mostly in file formats. The other was concerned with the electronic exchange (e-mail, file transfer) where the rules and formats that allowed an exchange between different platforms were the exception.

At the present time this situation is changing very rapidly. Multimedia mail allows for the exchange of structured information that may contain text, images, sound, and even active (executable) components. Exchange standards are appearing and are more and more included into offered multimedia e-mail software. However only marginal semantic structuring and layout capabilities are offered and for persistent storage purposes file systems are dominant.

Multimedia document manipulation also relies more and more on standards, e.g. SGML/HyTime, ODA/HyperODA, STEP/ Xpress, and on advanced modelling concepts that allow for spatial and temporal links. WWW uses "html" a variant of SGML but not yet extended with HyTime features. Again file systems are primarily used for document storage and concurrent or cooperative access remains problematic.

In this position statement we shall look at the requirements and possible solutions for archive systems that are able to handle multimedia documents and multimedia mail. Out of the many aspects we have selected the data modelling issues in the face of SGML/HyTime documents and the archiving concepts under the assumption that multimedia documents will have to be persistently stored for later retrieval and use in a widely distributed computer environment.

F2.2 Modelling and Storage of Multimedia Documents

At GMD-IPSI a publishing environment is developed. It provides the means to produce different electronic publications. A sample publication developed in this framework is an encyclopedic hypermedia application for art historians called "Dictionary of Arts" [Aberer, Boehm, Hueser 94]. We further envisage an open publishing application to provide a knowledge resource in conjunction with other tools and publications. The base of material comprises over 11,000 files occupying some 66 megabyte of storage. This represents approximately one quarter of the articles being prepared. For the electronic version there is considerable leeway for enhancing the interplay of text and factual representations with multimedia content. Examples are color images, graphics, video/audio annotations.

Most multimedia applications involve a diversity of conventional data types like numbers, text, and tables combined with media data like images (bitmaps), graphics, audio, video and animations. Especially, the users of a cooperative publishing environment - the overall research focus of our institute - need to be supported in storing, retrieving, and accessing large amounts of multimedia information in a distributed environment. Additionally, users should be able to control presentation of continuous media to allow more than the conventional consuming of e.g. video at television. Hence, most multimedia applications need shared stores which enable management of active media objects.

Such information comprises explicit modelling of relationships between the different kinds of content in general. Classification of content for the selection and presentation is also mandatory. At any time, the information contained in the database has to be consistent. The information source has to maintain constraints and rules for all kinds of entities involved in the production as well as constraints and rules on meaningful links between entities.

SGML document-type definitions can be seen as documents. We have designed a document-type definition whose instances are just document-type definitions. Usually we refer to it as ‘super-DTD'. Before a DTD, together with conformant documents, can be inserted into the database it must be transformed to an instance of the super-DTD. Then it can be inserted into the database the same way normal documents are (1). The classes representing the super-DTD in the database have been created within a bootstrapping mechanism when the database was created. The classes corresponding to the element-type definition of the DTD that has been in the previous step are generated (2). Now documents conformant to that DTD can be stored in the database (3).

F2.3 An Archiving Teleservice Using Mail

We introduce an archiving and retrieval teleservice for multimedia documents called Multimedia Archiving (MMA) [Timm, Rakow 94]. Connectionless, asynchronous multimedia mail is applied as a means for interchanging multimedia documents between archive clients and an active multimedia archive server. {FOOTENOTE: Another implementation of this teleservice uses the DFR standard as an access mechanism [Rakow et al. 94].} Since the special requirements of multimedia data within a networked environment are reflected in the underlying mail implementation as well as in the archiving components, it extends those functionality provided by usual electronic document archives. One feature is that searching for documents is possible by descriptive search criteria addressing document contents as well as multimedia specific data. This allows for example to select documents which do not contain video clips longer than 1 minute. Another feature is support for dynamic document composition. This allows for the users to retrieve other versions of originally archived multimedia documents which are dynamically created by the archive. In other words, a copy of the original document is tailored to meet the user's individual information needs, preferences, workstation environment and cost restrictions (teleservice providers charge, network and storage costs). Furthermore, it is allowed to retrieve only a description of a document's structure which can be helpful for a user who has to decide if a certain document should be retrieved completely. The uncertainty with respect to the returned amount of data faced in formulating queries is less critical. Users can specify the alternative - (1) a report of number of matching documents, (2) short description of document contents, or (3) complete documents - to be chosen by the archive server in dependence of the number of matches.

The architecture for implementing MMA by using mail is based on a client/server approach. The archive server and each client are directly connected to a Multimedia Mail User Agent (MMM-UA). Each MMM-UA has access to a X.400 Message Transfer Agent (MTA) and an External Reference Manager (XRM) in order to support synchronous access. In our prototypical implementation, the multimedia documents are accessed via the data manipulation language (DML) provided by the DBMS used for the realization of the archive server. Alternatively, a standardized language like SQL, or ‘‘mailadopted" standards for synchronous client/server access, e.g. DFR, or RDA, could also be supported (respectively those subsets of these standards that specify the query functionality). However, the deficiencies in currently available versions of such languages with respect to a realization of the required archiving functionality have to be compensated by more sophisticated client functionality. The provided descriptive access to the documents includes access to a set of application specific search attributes that serve as predefined search criteria for the client.

The Calendar of Events (CoE) is a sample application of the archiving teleservice. For example, an event description for a concert contains a picture showing a theatre stage, a video clip about the announced top actor, digitized newspaper critics, and an audio sequence presenting the latest song of the performing interpret. The descriptions can be inserted into an archive by information providers in a cooperative way. The initiator of an insertion specifies a workflow description which is handled and executed by the archiving system. Document parts are exchanged by sending mails to the persons involved. The events can be queried by Customers according to their interests and are presented at their workstations. It is planned to enable customer feedback like ticket-orders or reservations, reports or critics per voice annotations.

F2.4 Using Multimedia and Object-Oriented DBMSs

The key idea to satisfy the requirements on modelling and storage of multimedia documents is to rely on the functionality offered by database management systems (DBMSs) providing expressive modeling primitives. DBMSs allowing data-model extensions are particularly appealing. A general-purpose schema reflects the complex semantics of typed hypermedia objects. Hence, applications are freed from reimplementing these semantics.

Therefore, in order to qualify as a basis for a document storage system a DBMS must have certain features to cope with multimedia documents. First, the requirement that the semantics of hypermedia objects must be captured basically reduces the set of applicable DBMSs to object-oriented ones. DBMS standard "built-in features" that have been developed for the general case can be used without much specific implementation efforts: transaction management facilitates concurrent access without violating consistency, dealing with distribution likewise is standard with DBMSs, and query languages have been developed to facilitate descriptive data access. A concept for time-dependency and synchronized presentation of multimedia data must be integrated in the data description and the query language [Aberer, Klas 94]. Furthermore, presentations and control of presentations at the user's workstation requires a client server architecture, specific buffering concepts, and networks supporting continuous or isochronous transport protocols. The access to existing multimedia databases requires usage of integration techniques. There is a need for a tighter coupling between a DBMS and a mailing system. A possible solution is to strictly interchange the high-volume data via standardized mechanisms and to employ a synchronous transport protocol for the exchange of data between the DBMS and clients. For the DBMS, this would require support of a continuous data exchange mechanism for exchanging continuous data.

F2.5 Results and Perspectives

F2.5.1 Stable Knowledge

Using an object-oriented DBMS, a storage system for multimedia documents can be modelled perfectly. An important feature of such a system is dynamic document composition. To allow format, and quality transformations as well as continuous data extraction operations, the document content has to be stored within the database. This is best accomplished by a Multimedia DBMS that features multimedia datatypes. With a query language that supports method calls in query statements, the document composition functionality is accessible for the users.

Multimedia mail together with a DBMS-based archive for multimedia documents can be used to build a multimedia archiving teleservice. However, due to the mailing delay, this approach is not suitable for applications that do not allow unpredictable response times. Typical applications are subscription services, offer services, and work flow management applications. On the other hand, by using electronic mail, the archiving teleservice's availability for the users is decoupled from the server's availability since mails are buffered in the mail system. This allows the realization of an active archive, which can perform actions without being triggered by some user request. If the teleservice only allows to retrieve documents as originally archived, it can happen that some components of retrieved multimedia documents do not reflect the user's individual workstation environment, preferences with respect to structure, or contents of documents. To allow users to accomplish a minimum of such retrieved "mismatching" information functionality for dynamic document composition should be provided.

In order to allow a document exchange between existing authoring environments for multimedia documents and multimedia mail, common exchange formats like SGML/HyTime or HyperODA should be supported by mail.

F2.5.2 Remaining Problems

For the specification of dynamic document composition operations required by the user, currently it is not fully clear where these specification has to come from, and how it is made. An interactive specification seems to be applicable. The application of graphical kind of specification seems to be an adequate approach which is for further investigation.

Multimedia mailing bears some potential for multimedia enhanced work flow management. Audio and video annotations, for example, can be used to add some kind of informal interaction between participating users to todays workflow management paradigms which mostly do only support formal or semi-formal interaction. An adequate multimedia enhanced work flow management model has to be built.

To support the exchange of multimedia documents between authoring systems and multimedia mail, two approaches are possible: (1) mapping of standardized document formats into mail-internal document format, or (2) explicit support of standardized document formats by mail. Both approaches are for further investigation.

F2.5.3 Research Agenda

The objective of our research is the development of concepts needed for multimedia information systems. An environment is build up (using available systems wherever possible) where concepts can be shown, explored, and tested. Applications in the area of cooperative publishing environments are developed. The concepts and developments are evaluated against user requirements. The impact of advanced object-oriented database technology in hypermedia publishing applications is considered. Furthermore, the development of archives accessible by public broadband networks serves as a testbed for our research.

References

[Aberer, Boehm, Hueser 94] Karl Aberer, Klemens Boehm, Christoph Hueser: The Prospects of Publishing Using Advanced Database Concepts. Conference on Electronic Publishing, Document Manipulation and Typography, EP94, Darmstadt, Germany, 1994.

[Aberer, Klas 94] Karl Aberer, Wolfgang Klas: Supporting Temporal Multimedia Operations in Object-Oriented Database Management Systems. IEEE International Conference on Multimedia Computing and Systems, Boston, USA, May 1994.

[Rakow et al. 94] Thomas C. Rakow, Peter Dettling, Frank Moser, Bernhard Paul: Development of a Multimedia Archiving Teleservice using the DFR Standard. To be published in: Proc. 2nd International Workshop on Advanced Teleservices and High-Speed Communication Architectures, Heidelberg, Germany, Sept. 1994.

[Timm, Rakow 94] Heiko Thimm, Thomas C. Rakow: A DBMS-Based Multimedia Archiving Teleservice Incorporating Mail. Proceedings of the First International Conference on Application of Databases (ADB'94), Vadstena, Sweden, June 1994.

G Conferencing and Collaborative Computing

G1 Context Embedding and Reuse in Cooperative-Software Development

Max Mühlhäuser, Tom Rüdebusch, University of Karlsruhe, Germany

G1.1 Stable Knowledge

For a rough overview about the state of the art in collaborative computing, we will look at the major application classes one can find today, and on the way in which multimedia is used in the context of these applications.

G1.1.1 Application Classes

G1.1.1.1 Desktop Videoconferencing Systems (DTVC)

Videoconferencing has truly come to the desktop recently. Typical PC or Workstation based DTVC systems comprise audio and video connections, a dedicated shared whiteboard application (drawing/writing software with support for multiple users), and - less often - application sharing support.

‘Application sharing’ denotes techniques which allow to replicate the user interface of legacy software (“your favorite text processor / spreadsheet”), so that the software can be used simultaneously by all participants of a conference. The term ‘simultaneously’ is rather exaggerated in this context, since ‘floor passing’ (also: ‘chalk passing’) schemes are applied to determine which one of the users may actually interact with the software at a given time. Since the legacy applications (which were not designed to be used in a cooperative manner!) may remain unchanged, the term ‘collaboration-transparent applications’ is used, too.

DTVC systems have been developed for a wide range of platforms, both as to the desktops (DOS, OS/2, MacOS, UNIX …) and as to the networks (ISDN, Ethernet, ATM …). Recently, an emphasis has been put on user-friendly conference management for multi-party multi-role ("different *kinds* of participants") systems.

Human factors oriented projects have emphasized new device types, applying, e.g., semi-transparent mirrors and computerized large whiteboards ('lifeboards').

G1.1.1.2 Groupware

In contrast to 'application sharing' approaches, dedicated software for computer-supported cooperative work - groupware - is not 'collaboration-transparent' but 'collaboration-aware'.

Groupware can be categorized in different ways. For the remainder of this position statement, it is only important to distinguish the focus within 'collaborative computing' on either 'collaboration' (which is then just 'computer-augmented') or on 'computing' (in which case rather common software is augmented for cooperative use).

The first case, which might be called 'computer-augmented collaboration', is typically centered around a 'social' activity such as game-playing, discussion, negotiation, decision-making, etc. The goal of 'computer-augmented collaboration' is to facilitate such a social activity by means of its 'computerization' by adding, e.g., support for collaboration over distances (telecooperation) or support for organizing or memorizing (parts of) the social activities.

The second case might be denoted as 'collaboration-augmented computing'. Here, common computing tasks such as document or graphics editing are extended in order to accommodate multiple users.

G1.1.2 Multimedia Use

G1.1.2.1 Transparent

The establishment of audio and video connections between participants in a video conference is *the* predominant way of applying multimedia to collaborative computing. At a closer look, the use of the term multimedia in this context may be questioned, since a computer-supported coordinated use of multiple media can hardly be observed - at best, the audio and video channels are resynchronized at the sink nodes to provide lip-synchronous presentation (even this is *not* the case, e.g., in typical ISDN-based systems, where audio and video each occupy one or more 'line-switched' 64kbps B-channels and delays are considered small enough to make sink-side resynchronization unnecessary).

G1.1.3 Transient

Since video and audio information are captured → transmitted → presented → lost, the notion of ‘transient’ media can be applied. Typical persistent media on the contrary (which might come from, e.g., a multimedia database) are rarely used in todays DTVC and groupware systems.

G1.2 Remaining Problems

In the remainder, major flaws will be identified in state of the art collaborative computing as described above. They will be described in such a way that relevant research goals can immediately be derived.

G1.2.1 Little or Inflexible Customization to and Integration with the Context of Usage

There is hardly any support for building *customized embedded* software solutions that would use DTVC and groupware technology in the context of the specific activities undertaken in an application domain (at an enterprise). *Physically,* video-conferencing has come to the desktop computer as mentioned. *As to the usage,* the audio and video connections might almost as well be handled by a separate device next to the desktop computer (provided that ‘application sharing’ software were available in the computer network). Examples about how such customization and integration might look like are given below and in the context of other flaws.

(Note that in fact, the missing integration of computer-based telecommunication and of DTVC with other computer-based activities is *the reason* for which separate conferencing devices - in part still analog - still play a certain role and for which, e.g., almost everyone still considers a separate telephone set on its desk more practical than operating the phone from the computer - despite ***I*** SDN!!)

With respect to the groupware-type ‘computer-augmented collaboration’ (cf. section 1), there *is* a high level of customization to a specific kind of ‘social’ activity, as was argued; however, there is again usually *no cusomization* to the way in which this social activity is carried out in an organization. E.g., the decision support systems known from the research community might be further augmented considerably if they could be customized to the ‘common’ subjects of decisions and to the ‘common’ specific constraints which largely influence decisions in a given organization (e.g., manufacturing constraints which *regularly* constrain product design decisions) and if they could be integrated with the information sources in the network from which decision-relevant information could be (automatically) retrieved.

G1.2.2 Little Adaptation to Operational and User Context

Current DTVC systems are often dedicated to a specific networking technology (*either* Ethernet *or* public - e.g, ISDN - medium-speed networks *or* high-speed networks such as ATM). They can hardly be adapted in an intelligent way to user preferences, communication tariffs and the like.

G1.2.3 Very Low Level of Reuse and Development Support During Software Engineering

Since low-volume sales or even totally customer-specific software production have not been emphasized in the past for collaborative software, the problem of building every DTVC and groupware application from scratch has not yet been considered a predominant problem. But already today, one can complain that maybe hundreds of DTVC projects exist in the world, most of them re-inventing the wheel.

A few specific groupware development libraries (e.g., GroupKit) or even development environments (e.g., GroupIT) exist. They provide early approaches to both reuse and customization of DTVC/groupware development. They have, however, not left the academic stage yet. This is in part due to purely technical reasons (robustness, programming languages and platforms used, etc.). In part it is also due to the fact that within the flaws identified above, reuse and customization are approached, but integration and adaptation are not yet sufficiently covered (in particular, the integration of groupware with other software from the 'context of usage' is lacking, e.g., with mission-critical legacy software).

As to the development support, a sophisticated development environment would have to support adequate modeling and design of both groupware / DTVC applications *and* the encompassing overall software solution in which these applications are used. It would have to provide specific syntactic and semantic support throughout the software lifecycle (first steps done in GroupIT). As an example, the development environment should 'understand' groupware-specific operations (e.g., "join <user, group>") and check their correct usage during design and implementation.

G1.2.4 Lack of Persistent, Integrated and Perceived Use of Media

Media persistence, integration and perception can be considered three steps toward better use of media in collaborative computing.

The value of persistence can be observed by looking at asynchronous types of computer-based human interaction, such as simple email or computer-based voice mail. Using these interaction types, users tend to store interesting mails in folderd, preserving them for further use: excerpts from a mail message can, e.g., be used as part of the "minutes" of a decision-making process; in contrast, the audio and video conversations which lead to a decision are usually *not* preserved (unless it was known upfront that a piece of conversation would be needed *and* a DTVC system with capturing features was used). Technically, there is no obstacle for making all kinds of business conversation (via phone sets, answering machines, fax, DTVC even eye-to-eye) accessible to the computer and thus *potentially* persistent on digital storage. It should therefore be a goal to render the transient media persistent which are used for synchronous communication, and to integrate all kinds of telecommunication with the computer (which is possible today, but hardly done due to the lack of advantage). There is - of course - still too much *cost* associated with storing all transient information in computers and still too much *effort* needed for retrieving the *relevant* information (think, e.g., of having to go over the stored audio recording of a 5h meeting in order to find the place “where Bob committed to do the paperwork”). In this context, the further levels of sophistication of media use may help.

In a DTVC system which is well *integrated* with its context of use, the audio and video recordings could be structured according to the overall semantics of usage (step within the social activity performed; part of subject matter treated; role of a group member; relations to other media used such as, e.g., a ‘topic-list-graph’; relations to information retrieved via an embedded legacy software package; etc.). The system can then also guess ‘important’ elements of a social activity and restrict the recording and persistent storage to such important periods. Integration thus helps to reduce both the storage cost and the retrieval effort for persistent (formerly transient) media.

With the ongoing, breathtaking advancement in media *perception*, further reductions of the storage and retrieval ‘penalties’ for persistent conversation media may be achieved. There is promising research in video indexing (automatically identifying ‘scenes’ in a video stream with different content, automatically retrieving few ‘characteristic’ still images per scene), in face recognition (and object recognition in general), in wordspotting (finding keywords in an audio stream), in speech and handwriting recognition, etc. There is reason to believe that some less ambitious approaches to these domains (wordspotting and video indexing in particular) have lead or will soon lead to results which can be ‘engineered’ into customizable, reusable software solutions of practical value.

This development brings advantages in two levels. On one hand, media use as such can become more sophisticated (think of placing DTVC participants around a ‘virtual conference desk’ instead of placing each ‘talking head’ into a separate screen window). On the other hand, together with the above-mentioned media integration, storage and retrieval of persistent conversation media can become much more efficient *and* the sophistication of collaborative-computing can be further enhanced (think of, e.g., automatic wordspotting for words which are automatically retrieved from the context of usage into which a DTVC system is embedded).

G1.2.5 Lack of Media Transparency

In order to explain this term, we will use a comparable development in the domain of distributed systems: it may be considered a sign of maturity of distributed systems that advanced distributed-software engineering approaches tend to make the characteristics, even the existence, of a distributed system transparent to the user and to the application programmer: 'distribution transparency' is approached. E.g., with distributed object-oriented programming, it becomes an installation-time or even run-time decision how to break up a large application into pieces running on different network nodes. Of course, *total* transparency should not be enforced since at distinct points in an application, distribution may still be relevant at design or implementation time and for these cases, transparency is not desirable.

In the context of multimedia, however, we are far from introducing media transparency analogous to the distribution transparency above. E.g., for the development of DTVC and groupware applications, it might be interesting to be able to defer the decision about the communication media used to runtime (based on built-in selection criteria, not on forced user-actions). The design and implementation framework would in this case just 'talk' about the need of two parties to have a conversation. The consequences of the 'media transparency' requirement are *very drastic:* the elements of a whole modeling framework would have to concentrate on the real-world semantics and avoid notions about concrete media as long as possible in the software lifecycle. This is in drastic contrast to present multimedia development environments; e.g., for object-oriented approaches, the class libraries and modeling concepts are usually centered around 'media object' and 'connection topology element' class hierarchies.

Note that the requirement of media transparency can be extended to all kinds of multimedia applications.

In addition to the above, media transparency is of specific importance in the context of multimodal user interfaces (e.g., speech commands *plus* pointing devices used for identifying the operands) and of applications which must adapt to different types of peripherals. The latter is particularly important in the context of cooperative software since special devices (notepads, lifeboards, etc.) are more common in groupware scenarios then elsewhere (cf. Xerox PARC ubiquitous computing research).

G1.3 Research Agenda

The subjects to be addressed in a research agenda have been identified above and shall just be resumed here. They can be summarized as the search for an adequate *software technology* for the development of *customized groupware/DTVC solutions*, featuring:

- customization to and integration with the context of usage,
- adaptation to operational and user context,
- reuse and development support throughout the lifecycle
- persistent, integrated and perceived use of media.
- support for media transparency

As to the organizational background, the necessary practical experience about the right level and way of customization support can only be obtained if there is a lot of attention to the customers. In other words, there should be an attempt to carry out *true* application-oriented projects, where the (customized, integrated) applications should be targeted towards day-to-day use in mission-critical tasks of an organization.

Finally, although the author belongs to the advocates of data highways and high-speed communication, he believes that at least a part of the research should be carried out in mixed-speed infrastructures and should include the incorporation of traditional telecommunication means (e.g., via the integration of ISDN equipment).

G2 Conferencing and Collaborative Computing

Henning Schulzrinne, GMD-FOKUS, Germany

G2.1 Introduction

This paper attempts to provide a personal perspective on some of the problems, issues and possible directions for conferencing and collaborative computing. As with all such endeavors, the result is likely to be biased by the personal experiences and prejudices of the author. References are mostly pointers to recent related work, where more comprehensive discussions may be found.

Given the long history of efforts in both conferencing (from the telephony and computer side) and computer-supported cooperative work, the number of tools actually in regular and commercial use is not that large:

- electronic mail
- bulletin boards (Usenet news, Lotus Notes)
- screen sharing tools (e.g., ShowMe from SunSoft)
- text-based conferencing systems like Internet Relay Chat, multi-user dungeons (MUDs) or similar systems offered by commercial information providers like CompuServe or America Online
- telephone conferences
- conference room and roll-about video conferencing systems using MCUs for usually no more than three or four participants

In the last few years, it appears that almost every self-respecting computer science department and government-funded project has developed a multimedia-conferencing tool, particularly now that many workstations and personal computers and even some X terminals are capable of audio I/O and at least slow-scan video output. However, this increased activity has still tended to concentrate on small-scale videoconferencing, suitable for no more than a dozen or so participants. Sometimes, it appears that the functionality of these conferencing systems is guided by the limited experiences of the developers, consisting of seminars and developer's meetings taking place in dedicated meeting rooms. Video conferences meant two to four talking heads in rectangles on a screen. However, recently, there have been attempts to support the richness of human communication situations, such as unplanned hallway encounters, drop-in seminars, panel discussions, jury trials, lectures, pay-TV, and more. However, just mapping familiar communication patterns into conferencing systems may not be enough. The challenge is to progress beyond ‘‘horse-drawn carriage" stage of system building, making use the possibilities of workstation-based conferencing that cannot be mapped into physical conferencing situations.

G2.2 Audio and Video

Video conferencing is often portrayed as the ‘‘killer application" that is going to motivate everyone to go out and purchase whichever hot new networking or workstation technology is being proposed. However, both practical experience and formal experiments [1] seem to show that the use of video in conferencing applications is somewhat secondary. Video does not affect task performance; at least in technical discussions, it is often used to display viewgraphs (rather poorly) and to indicate how many people are still physically present within the conference. These tasks can also be filled by video with a few frames per second. For conferences with more than three or four participants, screen real-estate quickly runs out, particularly if other applications such as shared editors or drawing spaces are to be used. The ability to quickly resize individual images either manually or automatically, e.g., based on audio or drawing activity, can help. It may mean that video conferencing rooms with multiple, large high-resolution screens may continue to be necessary even after every workstation is video-equipped. The widespread use of video cameras at home seems to have made their presence less intimidating. On the other hand, good quality audio, with true full duplex communication and echo cancellation, possibly enhanced with spatial cues [2--4], appears to have been neglected. Good quality audio and video will prove particularly challenging in workstation-based conferencing since offices are often noisy, have acoustically reflective walls and lighting is inappropriate.

G2.3 Hardware Support

Currently, real-time multimedia services are provided both by general-purpose processors, without assistance without analog/digital conversion, and special-purpose add-in boards. It remains to be seen whether support for video and high-quality audio is closer to network services, where protocol processors have fallen out of favor, or graphics engines, where they have become standard in even the low-end

PCs. The expansion of video and high-quality audio data from encodings like JPEG or MPEG to pixel data argues for keeping it off any shared buses, and probably out of the memory system. (General RAM speeds have not increased significantly beyond the 70 ns mark.) Also, unlike network packets, the result of video decoding is usually not processed further, beyond moving it on the screen or simple scaling. On the other hand, it is far easier to introduce a new video coding if it can be distributed as software. For encoding, particularly motion estimation, only hardware-assisted solutions with low-level parallelism appear to be feasible. Work in developing video codecs suited for software encoding and decoding continues [5], but high coding gains seem to require motion estimation. Interoperation between dedicated circuit-switched codecs and workstation-based video conferencing is feasible with software-implementations of video coding standards like H.261. Desirable are also codecs that work reasonably well for both ‘‘natural" data like human faces and clothing as well as computer-generated material such as text on slides.

G2.4 Operating System Support

Neither standard multitasking nor real-time operating systems are well matched to the needs of real-time multimedia conferencing. Standard Unix scheduling favors I/O intensive processes, while software-based codecs may not get sufficient computational resources. Real-time operating systems provide the desirable means to lock processes into memory, but their rigid scheduling disciplines demand a detailed knowledge of the CPU requirements of the applications, something which usually neither application programmer nor user has. To make matters worse, the computational requirements often depend on a complex interaction of system resources (8-bit vs. 24-bit screen for dithering, say or system cache speed), user settings (frame rate and image size), encoding, image size and even program material. It has been proposed to make applications learn about their resource needs [6], but the success of that appears doubtful, given that for a video application, for example, CPU usage depends, among other factors, on the frame rates and number of sending sources, which may be changing constantly. Attempts to modify existing operating systems by adding a real-time scheduling class have caused system instabilities and had to be abandoned [7]. While ideally many real-time sources are periodic, network delays may cause bunching of packets. If CPU resources are made available only periodically, only parts of these packet bursts will be processed, leading to unacceptable quality degradation. Other real-time sources like slow-scan video may not be periodic at all. Overall, there remains work to be done to arrive at robust scheduling policies that guarantee sufficient resources to real-time applications without starving others and without having to have precise knowledge of application demands. Real-time applications may require kernel services, so that their scheduling has to be treated carefully. It may be necessary to either delegate compute-intensive real-time tasks to dedicated processors with low utilization.

While POSIX and X11 allow reasonably quick cross-platform development on Unix systems for many interactive applications, every vendor seems to have their own audio and video programming interface. Some attempts have been made at cross-platform APIs (NCD netaudio, DEC audiofile), but these appear to have a number of shortcomings. Most audio APIs seem designed mainly for playing back audio clips rather than real-time use. Netaudio and audiofile also abandon the Unix device-as-file model, requiring separate hooks into event handlers, separate read/write routines, and so on. This appears to be a step backwards and makes it difficult to use the same code to read from an interactive audio device and a file containing audio data. An audio API should allow sharing of the single speaker by several applications by either mixing at different volumes or priority override. External applications like VU meters or automatic gain control should be attachable to the audio input and output, without having to be replicated for every application. Separating audio source and sink by a network requires sophisticated playout adjustment at the receiver, particularly if the solution is to be usable beyond an uncongested local area network. Thus, a general system solution seems preferable to every application having to develop their own solution. Difficulties remain; for example, indicating the current talker is more complicated, as is the compensation for losses or other interventions by the application. Thus, either the audio library has to do perform almost all desirable audio services or very few beyond mixing, volume control and the like.

Also, for video, a clean separation of video frame grabbing and encodings appears difficult if extraneous normalizations and copies are to be avoided.

G2.5 Network Support and Protocols

Deployed networks are ill-suited for truly flexible computer conferencing. Synchronous networks like ISDN, switched 56k or higher-speed lines and the protocols for these networks like H.320 suffer from the difficulty of allocating bandwidth dynamically among a dynamic mix of applications. Multicast requires multipoint control units and is limited to fairly small group sizes. Framing protocols like H.221 are difficult to handle in software. Deployed packet networks like the Internet offer much more flexibility, but poor quality if even a small fraction of the overall bandwidth is used by real-time services. (The German X.25 network underlying the German research Internet seems particularly unsuited for real-time services, even though reliable transport protocols like TCP show acceptable performance.) Strategically placed users can, intentionally or not, flood the network with video traffic. More robust forms of multicast (rather than the current truncated broadcast) need to be developed and deployed, possibly with different protocols for sparse and dense groups. Diagnostic tools are just starting to appear. ST-II [8] has been proposed as an alternative solution, but it suffers from implementation complexity and poor scaling since the sender has to explicitly establish a connection to every end point. Using ST-II over ATM, both being connection-oriented, would appear to be a promising approach, but the two intertwined connection establishment phases add further delay and complication [9].

From the beginning, ATM was billed as a true multimedia service, with guaranteed quality of service. So far, commercially available switches offer at most a simple priority mechanism and thus cannot offer guarantees unless the combined peak rates of all VCs is less than capacity. (Clearly, any network technology can offer good service under those conditions.)

The debate over the proper role of ATM continues, with approaches ranging from treating ATM like adjustable-rate point-to-point links in a connectionless internetwork (classical IP over ATM [10]) to connectionless servers, i.e., yet another network-layer protocol, to dynamic connection establishment similar to the approaches using for routing IP over X.25 and ISDN, and finally pure ATM end-to-end solutions, possibly directly carrying MPEG streams, without any network layer protocol. Initial multicast support of switches and signaling protocols appears to be poor, with at best only homogeneous 1-to-N multicast in the offing. (Also, while theoretically supported, parallel connection establishment seems to be rather limited in scope, leading to extremely long connection-establishment times.) Receiver-initiated signaling as proposed for IP (RSVP, [11]), ST-II and ATM could relieve hosts from the burden of managing dynamically changing multicast groups of several hundred end systems. RSVP, in particular, offers an interesting approach to establishing state in datagram networks, but its error recovery properties and interaction with routing need to be explored.

At the transport level, all-in-one combined network/transport protocols like various derivatives of XTP [12] compete with simple special-purpose protocols like RTP [13] for the unreliable delivery of real-time data. The complexity, large header size and lack of routing support for protocols like XTP (without being encapsulated) have limited their use to demonstrations. RTP, based on packet-voice work dating back into the 1970s and more recent extensions to IP multicast by V. Jacobson, has also attempted to accommodate parties beyond the actual media agents, like quality-of-service monitors, recorders, firewalls, filters, bridges and the like. Synchronization between media streams has been the topic of a large body of research [14--16]. It appears, however, that once synchronized clocks are available with clock differences of a few milliseconds [17], the problem is largely solved. Explicit synchronization algorithms appear to provide special-case multi-party clock synchronization. Some further experimental work on playout synchronization in different networks is useful, although the need for sophisticated algorithms arises primarily in lower speed packet networks like the Internet [18].

The integration of resource discovery tools like the World-Wide Web with multimedia conferencing and collaborative computing is just beginning to be explored.

G2.6 Structure

Beyond providing for individual media sessions, a number of researchers have explored metaphors for conferences to provide structure and navigation. Examples are virtual meeting rooms [19,20] and hierarchical conferences, where a conference can itself be a member of a conference [21]. Also, operations on conferences as objects such as adding, merging [21] have been studied. It remains to be seen whether these models are flexible enough to encompass most real-life communication situations and whether the additional complexity is warranted and can be represented to the user in meaningful terms. The wholesale moving of participants as implied by conferences as objects may run counter to the desire of individual control on the part of participants.

G2.7 Conference Control

The scope of the term conference control is not defined precisely. It is generically applied to those aspects of computer-mediated communication, particularly synchronous ones, that are not concerned with data transport but rather with providing a structured, dynamic framework for a number of media sessions and a set of users and auxiliary tools. Conference control establishes agreement on common state (e.g., set of permissible audio encodings, the identity of a moderator or access rights (floorcontrol)), helps with adding new users to a session and reserves necessary network resources. Conference control may be layered, in that simple, per-media stream session management is utilized by a higher-layer protocol to tie together several stream. Clearly, conference control is related to signaling in the telephony world, particularly the notion of calls consisting of bearer services and aspects of negotiation.

Many conferencing systems (like other CSCW systems) had a rather idealized notion of how the ‘‘real world" works: a registration, admission and negotiation phase, followed by the conference proper, finally the closing. Even more so than in physical conferencing, however, it appears that electronic, particularly workstation-based conferences are far more fluid, with participants joining and dropping out of individual media sessions or the whole conference, taking phone calls, answering questions from people walking into their offices, etc.

Given a large number of participants and the complexity of applications and networks, it is likely that parts of a conference will misfunction during a session. Rather than terminating the conference, it will usually be more desirable to continue with as many participants as possible. The ‘‘sticky" conference control protocol [22] puts particular emphasis on such robustness. It appears that in many cases it is unrealistic to expect to have a universally agreed-upon state. Internet multimedia conferencing, exemplified by applications like vat, nv, nevot [23] and wb, take this to the extreme of simply periodically announcing the presence and state of participants via multicast. The announcement period is varied randomly to avoid synchronization, with increasing mean as the number of participants increases. For an audio conference with a media packet rate of 50 packets per second, about one hundred sites can be supported at an overhead of 1% and an awareness-lag of about one minute. For very large conferences of thousand or more participants, ‘‘selective awareness", indicating recent or signed-up speakers, may be needed, if for no other reason that displaying a list of a thousand participants on a screen is not very useful.

In this model, conferences consisting of several media sessions are announced through a multicast session directory (sd) by the initiator. This style of conference also takes the approach of controlling the delivery of data at the receiver rather than through, say, remote control. Similarly, access is limited by encryption and key distribution rather than an invitation process often found in small-scale conferences. This follows the basic philosophy of open networks that only encrypted data is truly safe from intruders.

Given the wide diversity of collaboration styles, it appears difficult to arrive at a canonical, all-encompassing conference control protocol or application. Rather more promising appear attempts to distill common, underlying control functions and allow combining these in appropriate ways. The agreement protocols [24,25] formalizes the problem of agreeing on shared state and modifying it based on voting rules. If eventual consistency is desired, some form of reliable multicast is necessary, an area where many protocols have been proposed [26--28]. For distributing state, reliable multicast has to deal gracefully with sites joining and leaving. It would be undesirable to halt a shared editor, say, if one of the participants is disconnected without formal connection termination. On the other hand, if two disconnected parts of a group continue to modify shared state independently, it may be impossible to arrive at a reasonable single shared state after the disconnection heals.

State agreement covers the functionality of floor and access control, media negotiation, moderator selection, and similar tasks. Other functional areas that need to be integrated into an overall architecture include directory services for sessions and invitation services. Other generic network services, such as encryption key distribution or user location services, are particularly valuable for computer-mediated communication.

G2.8 Tools and Applications

Collaborative computing will use both tools specially designed for use within conferences and generic tools. The latter is generally desirable as conferences are not the most opportune time to experiment with unfamiliar applications. Generic mechanisms for application sharing have been developed for X11 [29].

Even if special multi-user tools are used, the same tools should be usable for both small and large conferences. Having to switch tools because a small discussion turned into a seminar is rather undesirable. Unfortunately, this may be difficult to realize, as techniques like a fully meshed net of reliable transport connections do not scale, but are easier to program than reliable multicast. Similarly, tools should work both in high-bandwidth, low-loss, low-delay local area networks as in wide-area networks with the opposite properties, even if with reduced functionality or quality. This does impose a burden of having to compensate for much poorer network service. For some tools such as shared editors, integration of synchronous and asynchronous operation into a single tool is helpful. For others, it may be preferable to have ‘‘attachable" applications such as recorders and playback devices.

Sharing application at a relatively low level like a windowing system imposes the same user interface on all participants, regardless of personal preferences and local capabilities. Allowing for different user interfaces implies the rather difficult task of defining abstract operations and common data formats or state descriptions.

While traditional telephony can guarantee access limitations and adequate privacy by appropriate connection setup, many packet networks have no effective way from keeping users from sending or even listening to a particular packet stream. Thus, the only effective means of limiting distribution is to give the receiver control over what is rendered on the local workstation. Privacy has to be ensured by encryption. This receiver orientation is made easier if mixing, for example is performed at the end system rather than at a multipoint control unit. Similarly, coupling of several media, e.g., having the video display follow audio activity, is best accomplished at the end system since it can implement any desired policy. (Filtering unwanted data in the network, however, maybe more difficult.) IP multicast is naturally receiver-oriented, but connection-oriented protocols like ST-II or ATM still require signaling from the receiver to the sender, with appreciable delays. (See discussion of RSVP above.) Despite the basic notion of receiver control, mechanisms for voluntary remote control of applications, including senders, can be helpful.

In terms of applications, we are just gaining first experience with conferences (actually, mostly seminars, lectures and the like) in the Internet scaling from two to several hundred. Beyond quality problems due to networks without resource reservation, operational problems for more complicated conference scenarios remain to be addressed. Invitation and coordination of speakers are among the issues.

Experience suggests that for application writers, receiver-oriented multicast is much easier to implement than explicit participant lists. Also, the same is true for retransmitting state periodically rather than explicitly exchanging state with new arrivals, albeit at the cost of larger delays.

Given the diversity of conference control mechanism and the wide range of applications, a flexible means of tying together media applications, supporting tools (recorders and playback applications, calendars, etc.) and control applications (floor control, conference control, etc.) needs to be found. It may be nice if applications can be locally distributed among hosts, but still appear as belonging to a single user and be directed by a single conference controller. In Unix-like systems, either linking libraries or communicating processes may be used, but only the latter is usually configurable by the user. Traditional interprocess communication is not suitable since in many cases, there is no clear 'client' and 'server' relationship [30,31]. Also, the same kind of information, say, about new sessions or participants, may be of interest to a number of applications. This suggests multicast. However, local IP multicast requires a clear division of messages into classes by the sender, something which appears to be difficult for conference control. If more than one multicast address is used, a directory and address allocation service is needed. As an alternative, the Network Voice Terminal (NeVoT) is exploring the use of 'application-level' multicast, where a messages are forwarded to a central replicator called pmm (pattern-matching multicastor) containing regular-expression filters installed by applications. Applications connect to the replicator, which forwards messages to interested parties. Messages describe operations on objects such as conferences, media sessions, and the like. This approach allows to built conferencing applications without source modification or to attach new tools to existing conferencing systems. With any distributed scheme such as this, error reporting and security becomes more difficult. A generic operating system facility appears desirable; a first, related step in that direction is the SGI Irix facility that notifies applications of changes to selected files.

References

[1] S. Gale, ‘‘Adding audio and video to an office environment," in Studies in computer supported cooperative work (J. M. Bowers and S. D. Benford, eds.), Human Factors in Information Technology, pp. 49--62, Amsterdam: North-Holland, 1991.

[2] T. Becker, ‘‘Konzeption und realisierung eines ‘‘virtuellen konferenztisches"," Master's thesis, Technische Universit"at Berlin, Berlin, Germany, Dec. 1993. Studienarbeit.

[3] S. Hayashi, ‘‘Increase in binaural articulation score by simulated localization using head-related transfer function," IEICE Transactions on Fundamentals, vol. E75-A, pp. 149--154, Feb. 1992.

[4] S. Masaki, T. Arikawa, H. Ichihara, M. Tanbara, and K. Shimamura, ‘‘A promising groupware system for broadband ISDN: PMTC," ACM Computer Communication Review, vol. 22, pp. 55--56, Mar. 1992.

[5] R. Frederick, ‘‘Experiences with real-time software video compression," in Sixth International Workshop on Packet Video, pp. --, Sept. 1994.

[6] M. B. Jones, ‘‘Adaptive real-time resource management supporting modular composition of digital multimedia services," in Proceedings of the 4th International Workshop on Network and Operating System Support for Digital Audio and Video, (Lancaster, U.K.), pp. 11--18, Lancaster University, Nov. 1993.

[7] J. Neih, J. G. Hanko, J. D. Northcutt, and G. A. Wall, ‘‘SVR4 UNIX scheduler unacceptable for multimedia applications," in Proceedings of the 4th International Workshop on Network and Operating System Support for Digital Audio and Video, (Lancaster, U.K.), pp. 35--47, Lancaster University, Nov. 1993.

[8] C. Topolcic, ‘‘Experimental internet stream protocol, version 2 (ST-II)," Request for Comments (Experimental) RFC 1190, Internet Engineering Task Force, Oct. 1990.

[9] O. Hagsand and S. Pink, ‘‘ATM as a link in an ST-2 internet," in Proceedings of the 4th International Workshop on Network and Operating System Support for Digital Audio and Video, (Lancaster, U.K.), pp. 189--198, Lancaster University, Nov. 1993.

[10] M. Laubach, ‘‘Classical IP and ARP over ATM," Request for Comments (Proposed Standard) RFC 1577, Internet Engineering Task Force, Jan. 1994.

[11] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala, ‘‘Rsvp: a new resource ReSerVation protocol," IEEE Network, vol. 7, pp. 8--18, Sept. 1993.

[12] B. Metzler and I. Miloucheva, ‘‘Specification of the broadband transport protocol XTPX." R2060/TUB/CIO/DS/P/001/b2, Feb. 1993.

[13] H. Schulzrinne and S. Casner, ‘‘A transport protocol for real-time applications." Internet draft (work-in-progress) draft-ietf-avt-rtp-*.txt, Sept. 1993.

[14] S. Ramanathan and V. P. Rangan, ‘‘Adaptive feedback techniques for synchronized multimedia retrieval over integrated networks," IEEE/ACM Transactions on Networking, vol. 1, pp. 246--260, Apr. 1993.

[15] D. C. A. Bulterman, ‘‘Synchronization of multi-sourced multimedia data for heterogeneous target systems," in Third International Workshop on network and operating system support for digital audio and video, (San Diego, California), pp. 110--120, IEEE Communications Society, Nov. 1992.

[16] T. D. C. Little, A. Ghafoor, C. Y. R. Chen, C. S. Chang, and P. B. Berra, ‘‘Multimedia synchronization," The Quarterly Bulletin of the IEEE Computer Society Technical Committe on Data Engineering, vol. 14, pp. 26--35, Sept. 1991.

[17] D. L. Mills, ‘‘Internet time synchronization: the network time protocol," IEEE Transactions on Communications, vol. 39, pp. 1482--1493, Oct. 1991.

[18] R. Ramjee, J. Kurose, D. Towsley, and H. Schulzrinne, ‘‘Adaptive playout mechanisms for packetized audio applications in widearea networks," in Proceedings of the Conference on Computer Communications (IEEE Infocom), (Toronto, Canada), June 1994.

[19] S. R. Ahuja and J. R. Ensor, ‘‘Call and connection management: making desktop conferencing systems a real service," ACM Computer Communication Review, vol. 22, pp. 10--11, Mar. 1992.

[20] T. O'Grady and S. Greenberg, ‘‘A groupware environment for complete meetings," in Proceedings CHI'94, pp. --, 1994.

[21] H. M. Vin and P. V. Rangan, ‘‘System support for computer mediated multimedia collaborations," in Proceedings of the 1992 ACM Conference on Computer Supported Cooperative Work (CSCW'92), (Toronto, Canada), pp. 203--209, ACM, Nov. 1992.

[22] C. Elliott, ‘‘A 'sticky' conference control protocol," Internetworking: Research and Experience, vol. 5, pp. --, 1994.

[23] H. Schulzrinne, ‘‘Voice communication across the Internet: A network voice terminal," Technical Report TR 92-50, Dept. of Computer Science, University of Massachusetts, Amherst, Massachusetts, July 1992.

[24] S. Shenker and A. Weinrib, ‘‘Managing shared ephemeral teleconferencing state: policy and mechanism." memorandum, Mar. 1994.

[25] B. Rajagopalan, ‘‘Consensus and control in wide-area group communication." unpublished memorandum, Nov. 1993.

[26] M. Handley and I. Wakeman, ‘‘Cccp: Conference control channel protocol -- a scalable base for building conference control applications." V1.4, Mar. 1994.

[27] L. L. Peterson, N. C. Bucholz, and R. D. Schlicting, ‘‘Preserving and using context information in interprocess communication," ACM Trans. Computer Systems, vol. 7, pp. 217--246, Aug. 1989.

[28] R. Aiello, E. Pagani, and G. P. Rossi, ‘‘Causal ordering in reliable group communications," in SIGCOMM Symposium on Communications Architectures and Protocols (D. P. Sidhu, ed.), (San Francisco, California), pp. 106--115, ACM, Sept. 1993. also in Computer Communication Review 23 (4), Oct. 1992.

[29] H. Abdel-Wahab and K. Jeffay, ‘‘Issues, problems and solutions in sharing X clients on multiple displays," Internetworking: Research and Experience, vol. 5, pp. 1--15, Jan. 1994.

[30] J. Crowcroft, ‘‘Remote procedure call: not a panacea for distributed computing problems." University College London, Feb. 1993.

[31] M. Roseman and S. Greenberg, ‘‘Building flexible groupware through open protocols," in Proceedings COSC'93, pp. --, ACM, 1993.

H Multimedia User Interfaces

H1 Seamless Multimedia Integration for Digital Libraries

Edward Fox, Virginia Tech, USA

H1.1 Introduction

Multimedia systems should have usable interfaces that allow their users to efficiently and easily carry out desired tasks. Those interfaces should be scalable, allow media integration, and be dynamic. Developing such interfaces in the general case is a large, varied, and difficult undertaking; therefore this paper only occasionally touches on the general problem. Primarily, then, we attempt to reduce the problem to manageable size by drawing examples and focusing on an important class of multimedia systems and the corresponding set of matching tasks: those relating to digital libraries.

In this paper, Digital Library (DL) refers to a complex that includes: a community of users, a large body of multimedia content, and the hardware and software systems that provide proper support to those users. Since DLs will have: a diverse group of users, an accumulation of multimedia human knowledge represented in any forms suitable for storage and redistribution, and a large collection of equipment and software, having suitable interfaces is a monumental challenge and important area of research.

H1.2 Stable Knowledge

Today's interfaces to library information are primarily focused on handling text and some images. We know how to do fairly well at indexing, searching, browsing, and hypertext usage (though we are still learning about authoring and adaptation to varying applications). As shown in the Envision interface, we have methods for managing and visualizing result sets, and these seem of particular value. Clearly, users want control, need to have low-tech solutions, and should have good tools to work with.

Today's interfaces to multimedia are based largely upon the technology developed by the entertainment industry: theme-park rides, magazines, movies, televisions, CD/stereo systems and speakers, TV/VCR/game control units, and MIDI components. On the other hand, interfaces to VR have been developed initially for defense and related activities, with expensive head-mounted displays, eye-tracking, gloves, and the like --- usually dealing primarily with graphics information.

The current state of such interfaces is that solutions affordable by users of DLs are usually focused (at any one time) on one or a few types of (related) multimedia data, such as graphics, music, audio, text, drawings, images, animations, or video. The baseline for DLs are the conventional forms of data found in libraries, such as books, magazines, newspapers, maps, paintings, slides, photographs, videotapes, and CDs. Most of these can be accommodated, though we are hard pressed to deal (with sufficient quality of service) with large maps or paintings, or high definition forms of video. That is, we can store and play back, albeit slowly, each type of media, separately, on special devices.

For displayable multimedia information, a common approach is to use a special viewer (e.g., xtiff, xv, mpeg_player, ghostview, Acrobat) that is constrained to a window for each type of object. Current prototypes of digital libraries adopt this philosophy. Mosaic launches general purpose viewers for each separate type. Hyper-G has a more sophisticated approach, providing special viewers (a la Intermedia) so that linking can be into objects in each media form. Further, Hyper-G introduces cluster (present at once) and collection (choose from a set) objects to help organize related parts of the presentation. The Trellis system goes further yet, using Petri nets to specify complex presentation and interaction sequences. In the various systems employed for the TULIP project, where page images are the sole type of data recorded, a single viewer suffices. In the CORE project, having separate viewers for different types (e.g., text, figures), has been shown to yield lower performance for chemists on tested tasks. This suggests that integration is a desirable goal.

Recently, in work at UCI, MIT Media Lab, Bellcore, and other places, there been real successes in developing integrated multimedia presentations. While there have been some initial successes, further work is needed.

H1.3 Remaining Problems

Thus, the first problem is integration. This relates to many areas of study, including: authoring, aesthetics, constraint-satisfaction, layout, document structure, hypermedia, etc. What styles of integration are pleasing, and which help in various task setting, for various types of users? What guidelines can be developed that are appropriate for human perception, cognition and focus? How can we identify and satisfy the many constraints that exist due to the multimedia information, presentation devices, computer and network capabilities, user preferences, and task requirements? How can authors be given sufficient guidance, proper tools, and yet be free to be creative in authoring applications and presentations? How does integration relate to linking in hypermedia systems? How can such link information, and history records of prior access, allow pre-fetching to improve performance? What representations should be used (e.g., according to HyTime, TEI, MHEG, PREMO) to describe relationships, interactions, and presentation semantics? How can layout in 3-dimensions and time be viewed, reviewed, summarized, and specified? How can the needs of authors, editors, reviewers, producers, viewers, annotators, and collaborators all be accommodated? In the case of DLs, what new types of collections beyond those commonly found today will become particularly popular, or used for training, education, presentations, research, and other activities? How will DL patrons differ in their preferences and needs regarding integration? In particular, what types of accommodations should be made for those with particular types of physical challenges?

The second key problem is scalability -- seamlessly looking at images or video in small or large forms (e.g., as we might through a telescope, or by turning around on a clear day from atop a mountain). This relates to resolution and screen size (e.g., wristwatches or desks or walls full of display). There has been work at Bellcore on infinitely scalable paper, relating to two-dimensional objects. Yet all these techniques wreck havoc with bitmap or pixel array representations of images, or other results from sampling that fixes an upper bound on the detail that can be perceived. Do we need to have true scalability of resolution, and if so, what methods give affordable solutions? Virtual reality, such as being explored with Hyper-G for information collection browsing, seems to be relevant, with metaphors such as objects (e.g., buildings) atop objects, to arbitrary levels of nesting. Yet, this is typically limited to synthetically generated graphics scenes --- can the ideas be extended? Further, how does scalability work in terms of volumes of information, numbers of users, and amount of usage --- will our networks and computers be able to handle the enormous load of interfaces making such demands against petabyte-scale DLs?

A third problem is indexing and search. In large DLs, we need to be able to index any object, regardless of its media, and to search through large collections based on (constantly changing selections of) attributes and their values, individually as well as in combination. Our interfaces must allow us to describe or give examples of objects in natural ways, and to rapidly browse and filter results. The Envision system supports query handling, retrieval and results management, but only based on descriptions using text or a small set of features. How do we index and search against images, audio, video archives? How do we extend current protocols for client / server interaction to support such large objects, index into their components, and only transmit components of objects? How do we cluster and associate related objects or sub-objects across media types? How do we visualize the concepts present, in abstract and media-dependent forms? How do we browse through them, organize and re-organize them, and specify queries using them as well as the base objects (and links) themselves? How do we search against different types of structure, looking for content in varying contexts? What composites are useful, like phrases are to text, in other media forms? How do we apply and perhaps generalize knowledge representation schemes like semantic networks and frames to better support concept management and document (in the general sense) indexing?

A fifth problem relates to usability testing and methods of interface development. Developing good GUI interfaces and testing them is a complex task; how much more complex is the development of good multimedia interfaces? How can we reduce the number of parameters and variables, and be guided during the formative evaluation process? How can we ever hope to undertake summative evaluations with large-scale DLs? Even focusing on evaluating toolkits that will be used for constructing multimedia interfaces (and we DO need such toolkits, and to refine them, since in general they will be used in the development of most applications), how can we make this into a manageable problem?

A sixth problem relates to training and help. We will, no doubt, devise totally new types of interfaces to DLs and for other multimedia applications. Will some individuals have considerable advantage over others in this new situation? Is not that already the case with pilots, astronauts, and others selected based on performance with complex simulators? Can the multimedia media complement and compensate so that a wider class of users can be accommodated -- much as students learning to read can be diagnosed and trained according to their mix of talents? Can we systematize this process, so we do not fall prey to the lowest-common denominator syndrome so common in current television programming? Can we develop novel help methods to help individuals even more?

A seventh problem relates to metaphors. The power of many interfaces is drawn from the metaphors adopted, but we must at the same time avoid inappropriate attempts to metaphor us in. What metaphors are natural for DLs? Can VR really be applied here in a useful way? How can we trade off the use of metaphors common to physical libraries, that may be only useful for transitional stages, with the emergence of new metaphors that relate to the content of DL objects? That is, how can we transition to using computer science metaphors for CS literature, and GIS-related metaphors for studying geography? At the same time, how do we avoid confusion, when readers wish to browse through heterogeneous collections? Or when we move to community networking situations, where other people in our neighborhood, or experts, or teachers, or colleagues, all play a role?

Clearly, there are numerous research challenges facing us in the construction of new multimedia and VR interfaces!

H1.4 Research Agenda

How then do we develop a research agenda? Here a few key points are in order.

First, we should focus on current and future needs before worrying at great length with capture and conversion of existing analog representations. A serious mistake in the DL arena, for example, is to worry about scanning page images and solving the document analysis and OCR problem, before agreeing on SGML DTDs, applications of HyTime, and other crucial representation problems that affect current publishing efforts. In the case of multimedia, it is crucial to think how best to undertake education, training, and other important tasks, and not to be limited by current approaches as we develop new approaches.

Second, it is essential to develop toolkits. One of the great problems with CD-I was that there were no adequate toolkits to be used by publishers. Not only do we need toolkits for publishers, but also for authors, artists, educators, presenters, game developers, and scientists involved in visualization. For medical, simulation, training, and other key applications, there should be specialized toolkits. Perhaps they will all make use of basic findings, a moderate-size set of metaphors, and have common foundations, but it is clear that there will have to be a number of different toolkits nonetheless.

Third, as a result of efforts to build large DLs, and from other similar efforts, there should be a concerted effort to have standard collections that can be used for scientific comparisons. In the information retrieval world, the availability of test collections, such as the recently developed TREC collections, has played a key role in making the field into a successful empirical science. Similar efforts, though unpopular because of the time and expense involved, should nonetheless be done for the multimedia field.

Fourth, interfaces should make explicit all distinctions that are important, and hide others. Key to hypermedia is the explicit handling of nodes, anchors and links. Key to information retrieval is the explicit handling of documents on the one hand, and terms/phrases/concepts on the other. Key to help systems is the handling of user concerns and questions, in context, but still at a meta level. Key to data handling is the separation of data and metadata when that is appropriate, and the integrated handling of them when different types of analysis are called for. Key to use by children (as studied by Borgman, and the Dannish group working on the Book-House), is making explicit the concepts and representations they find appealing.

Fifth, we must address the many problems discussed earlier in a coordinated fashion. Interface studies should go hand-in-hand with other multimedia research, else we will later have to re-visit the other research efforts (as we have done in the

Envision project, where the deficiencies of the protocols used for WAIS and WWW became readily apparent as we developed a sophisticated task-oriented interface). Scalability of resolution has tremendous effect on compression methods, networking requirements, and storage approaches. Scalability of usage and volumes of information should be considered through work-load monitoring, system modeling, and simulation studies. Integration by its very nature forces concern with coordination of efforts, since it calls for work in a wide variety of areas to achieve a common, elegant, and usable result. Ultimately, this should rest upon secure formalisms, but we have yet to develop those to the degree now required.

Finally, it should be said that since we are at an early stage in these efforts, we must be flexible, not be bound too tightly by standards, must try to do excellent work in specific situations (e.g., certain types of DLs), and work closely with others exploring the use of multimedia systems to support important human activities.