José Encarnação, James Foley,
Ralf Guido Herrtwich (editors):

Fundamentals and Perspectives of Multimedia Systems

Dagstuhl-Seminar-Report; 92
04.07.-08.07.94 (9427)

ISSN 0940-1121

Das Internationale Begegnungs- und Forschungszentrum für Informatik (IBFI) ist eine gemeinnützige GmbH. Sie veranstaltet regelmäßig wissenschaftliche Seminare, welche nach Antrag der Tagungsleiter und Begutachtung durch das wissenschaftliche Direktorium mit persönlich eingeladenen Gästen durchgeführt werden.

Bezugsadresse: Geschäftsstelle Schloss Dagstuhl
Universität des Saarlandes
Postfach 15 11 50
D-66041 Saarbrücken, Germany
Tel.: +49 -681 - 302 - 4396
Fax: +49 -681 - 302 - 4397
e-mail: office@dag.uni-sb.de
www: http://www.dag.uni-sb.de/dagstuhl.html

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. 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. Encarnacao

Table of Contents

1	Multimedia Systems 1
1.1	Survey (White Paper Presentation by Ralf Herrtwich) 1
1.2	Position Statements 2
1.3	Research Items 5
2	Media Encoding and Compression 7
2.1	Survey (White Paper Presentation by Ralf Steinmetz) 7
2.2	Position Statements 9
2.3	Research Items 13
3	Abstractions for Multimedia Computing 15
3.1	Survey (White Paper Presentation by Thomas D. Little) 15
3.2	Position Statements 16
3.3	Research Items 19
4	Multimedia Storage and Databases 23
4.1	Survey (White Paper Presentation by Desai Narasimhalu) 23
4.2	Position Statements 25
4.3	Research Items 26
5	Multimedia Networking and Communication 31
5.1	Survey (White Paper Presentation by Fouad Tobagi) 31
5.2	Position Statements 32
5.3	Research Items 36
6	Multimedia Documents and Mailing 37
6.1	Survey (White Paper Presentation by Gerd Schürmann) 37
6.2	Position Statements 40
6.3	Research Items 43
7	Conferencing and Collaborative Computing 45
7.1	Introduction (White Paper Presentation by Eve M. Schooler) 45

7.2 Issues on Widespread Telecollaboration 47
7.3 Conferencing as a Vertical Application 51
7.4 Context Embedding and Reuse in Cooperative-Software Development 54
7.5 Research Agenda 55

8 Multimedia Interfaces 59
8.1 Survey (White Paper Presentation by Steven K. Feiner) 59
8.2 Position Statements 59

1 Multimedia Systems

Darin E. Krasle, Georgia Institute of Technology, Atlanta
Thomas Käppner, IBM, Heidelberg

1.1 Survey (White Paper Presentation by Ralf Herrtwich)

Multimedia is a term which is not well defined, but which has become popular in describing the capabilities of the emerging generation of hardware and software systems. The implication of the word is that several means of communication, or media, are supported by the system under consideration. Audio, text, and graphics are examples of media.

From a human perceptual standpoint, each medium may be classified according to its presentation values and presentation space. The presentation space for various media take various forms such as physical space for graphics or acoustic space for audio. The presentation values vary similarly in that text uses sequences of letters, audio a series of pressure waves. Each presentation space has presentation dimensions, such as the physical dimensions on a computer screen. Time is a dimension which is contained in all presentation spaces and may be perceived as discrete or continuous.

The processing of continuous media such as video and audio poses a great challenge to computing, but also enables users to manipulate these media in ways that were formerly unavailable. This added power allows more powerful interactive systems to be constructed which may replace traditional technologies and devices.

As the migration from text-based interaction to graphical user interfaces has shown, new paradigms are needed to effectively utilize new technology. Applications for multimedia are quickly penetrating a wide range of markets including office, home, shopping, learning, and entertainment. There is no doubt that as the enabling technologies for multimedia improve, more applications will become feasible.

The technologies of inexpensive bit-mapped displays, digital signal processing, and optical and magneto-optical storage have already brought multimedia a long way and continue to be promising. With the advent of better means of broadband communication such as Integrated Services Digital Network (ISDN) and Fiber Distributed Data Interface (FDDI), along with better protocols such as Asynchronous Transfer Mode (ATM), the feasibility of computer-based multimedia communication is greatly enhanced. Better software support in terms of standardization is also becoming more important in making multimedia applications more independent from underlying technologies.

1.2 Position Statements

1.2.1 Daniel C. Swinehart

This position statement attempts to identify some of the problems that hinder development in the field of multimedia systems and appeals to the community to take the first steps to improve the situation.

This position is stated as a set of concerns about valuing specific multimedia applications over the basic system components that must be perfected in order to support future applications. A summary of these concerns is that "we will:"

- produce an isolated "multimedia systems" community that is largely unaware of progress in the development of the individual components that comprise the 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.

Based on these considerations, the focus turned to proposed goals for the Dagstuhl workshop. Since the premise was that the field is largely more in need of taxonomy and organization than further innovation, the workshop should be utilized to produce a volume that could be used as a basis for sharing state and understanding. This suggests examining the topics of each session with specific questions in mind:

- Is There Much More Work to be Done? This question will identify those areas where significant additional work is needed before reaching any sort of maturity. In these areas, it is important to encourage development.
- Is It Time to Choose? Areas that are quite well understood should be identified in order to select a small number of candidates to promote as standards to the rest of the community.

At a bare minimum, the workshop should produce a list of solutions used in multimedia systems today serving as a useful taxonomy. It could contain information about which facilities are experimental, which commercial, which publicly announced but proprietary held, and which openly available.

From a component perspective, the most important issue concerns user interfaces; special attention should be given to reduce their quantity and increase their quality.

The discussion was opened with a component-oriented theme stressing the importance of considering factors such as flexibility, combinations, reuse, usefulness, interfaces, standards, and robustness when examining the components of multimedia systems. During the discussion, the audience expressed concerns about the nature of the motivations behind the design of multimedia systems. It was pointed out that users' needs must be identified and that both technology push and user demand pull should affect design and development. (The speaker doesn't necessarily fully agree with this view; the value of some components is by now well enough understood that they can be well-specified based on what we know already).

Designing everything up-front was considered to be a bad approach, and, as the speaker indicated, the tendency among developers is a bias toward specific functionality which may not adequately address users' more general needs. He went further to say that the pressures to bring products to market, coupled with the ambitious nature of product development projects, make it difficult to examine the cognitive aspects of designs in great depth. It is infeasible to have cognitive scientists knead over every design without greatly affecting the development process. He stressed the importance of concentrating on creating high quality, integrated infrastructure as well as integrated applications.

1.2.2 Stephen Casner

In this position statement the Internet is presented as a key technology for multimedia, both influencing the purpose and structure of multimedia systems and changing to meet their requirements. The history and current demands of multimedia traffic in the Internet are highlighted and the research is summarized that is required to support them.

The hope is expressed that the flexible communication enabled by the Internet will make more sources available than just interactive home shopping and old movies, thus adding options to the purposes of multimedia systems. The Internet communication model as an example of the notion of smart end-systems, as opposed to the dumb terminals of telephony, fits well with the author's view of an appropriate structure of multimedia systems.

The history of real-time media on the Internet extends from early packet voice experiments, to resource reservation and multicast in the Stream Protocol (ST) and ST-II, to Internet Protocol (IP) Multicast and its deployment in the Multicast Backbone (MBONE) which is a virtual network layered on top of portions of the physical Internet providing the IP multicast connectivity. The MBONE has been used to carry live audio and video data from a wide variety of events such as Internet Engineering Task Force (IETF) meetings with a remote audience of about 600 participants. The MBONE has grown to over 1000 networks and subnets in 20 countries, thus proving the demand for multimedia communication with worldwide span. Communication has even been bidirectional in that every participant is not only able to watch and listen, but can also speak back to the assembled group. Through the use of packet switching in combination with silence detection, the overhead of such return channels has been quite low.

Since IP multicast is currently being implemented in IP production routers it will become a standard network service over the next few years. At that time the full network bandwidth will be available to multicast traffic which could easily overload the network with real-time audio and video if there is no mechanism to control its use. Furthermore, the best-effort delivery service that has always been acceptable 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 these 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. The cost of the additional capacity must be recovered, however, 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 as opposed to 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.

Applications that utilize wide-area distribution on a very large scale require distributing 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. Sources may still need to be in control of charging, by, for example, restricting access via encryption, but that work can also be distributed so that receivers need not interact directly with the source node.

The discussion was begun by introducing a list of key technologies. It was asserted that hardware will be supplied, but that networking and communication support are lacking. The facsimile machine (FAX) was proposed as an example of a successful melting of technologies and support. It was pointed out that FAX machines, in contrast to multimedia authoring tools, have simple interfaces and are easy to use. The adherence to standards, affordable cost, ease of use, and proper market deployment were cited as factors which led to FAX success and that these areas should be addressed when developing multimedia systems.

It was also pointed out that future systems may vary greatly from the way the Internet is currently set up, and that this would be possible using current IP network technologies. Minitel and financial networks were cited as examples of successful existing networks with architectures different from that of the Internet. It was claimed that no existing system solves the problems involved in scheduling, and that standards are needed not only for protocols, but also for switching. A distinction was drawn between transmission protocols and control protocols, the latter of which will become increasingly more important in solving problems dealing with billing and quality of service.

1.3 Research Items

Each of the speakers indicated a number of areas in which research would prove productive to the field of multimedia systems. Important questions that need to be answered include the following:

- What architecture should multimedia systems have? In the past real-time environments have been proposed to support the processing demands of multimedia data on workstations. Will the advent of more powerful processors make these demands less critical allowing real-time support to reside at more superficial levels? It is possible to design multimedia architectures that adapt the requirements of processing multimedia information to the capabilities of the system. Would such systems be more successful than those guaranteeing real-time service?
- Which reusable components/system layers/modules can be identified for multimedia systems? An examination of multimedia systems as they are built today could serve as a starting point to identify common functionality. Important factors for such functional modules include flexibility, combinability, reuse, usefulness, interfaces, and robustness. Such a taxonomy of multimedia systems could eventually lead to standards for a multimedia system architecture.
- What are architectural approaches that scale well? As experience on the Multicast Backbone has already shown, multimedia systems are widely available which increases the demand for system architectures that scale well to a large number of participants. Those architectures must minimize the overhead of administration and incorporate billing models for the transmission of real-time media.
- What are the new paradigms for the use of audiovisual information in the computer context? Since various media can convey information in different presentation spaces and with different presentation values, new paradigms must be developed to determine which combinations lead to effective and optimal presentations. The role of the user must be considered. Passive users have different presentation needs than interactive users. The flow of information in each direction must be thoughtfully considered in order to ease interaction between system and user. Multimedia systems give people the opportunity to control information in ways that were formerly unavailable. What needs to be determined is exactly how people can manipulate this information and what kinds of mental models are

necessary to allow people to operate in such an environment. What new tools and methods will prove useful and necessary?

- How should user interfaces and authoring tools be defined? User interfaces and authoring tools might be considered as enabling technologies for the production and consumption of the actual data, not just its transport. Producing high quality multimedia documents requires artistic composition skills and often a cyclic development process. Good authoring tools and interfaces are needed to support the flow of the creative process and allow refinement by providing facilities for previewing and making modifications. User interfaces will ultimately determine any system's effective functionality since functions that are difficult for users to access will seldom be used.

2 Media Encoding and Compression

Rüdiger Strack, Fraunhofer-Gesellschaft, Darmstadt

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) Darmstadt
Maria Perez-Luque, Boston University

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.

4 Multimedia Storage and Databases

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

4.1 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.

4.2 Position Statements

4.2.1 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.

4.2.2 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.

4.3 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 applica-

tions. 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, New York

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.

6 Multimedia Documents and Mailing

Thomas Meyer-Boudnik, University of Mannheim
Blair MacIntyre, Columbia University, New York

6.1 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.

6.2 Position Statements

6.2.1 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.

6.2.2 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.

6.3 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 workflow 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?

7 Conferencing and Collaborative Computing

Michael Altenhofen, Digital Equipment, Karlsruhe

7.1 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:

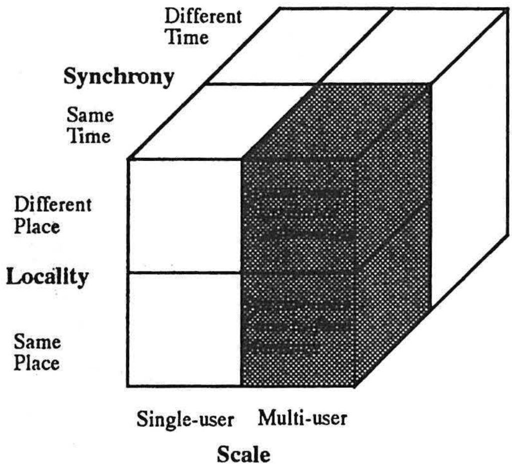

Figure 1: Collaboration Matrix.

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.

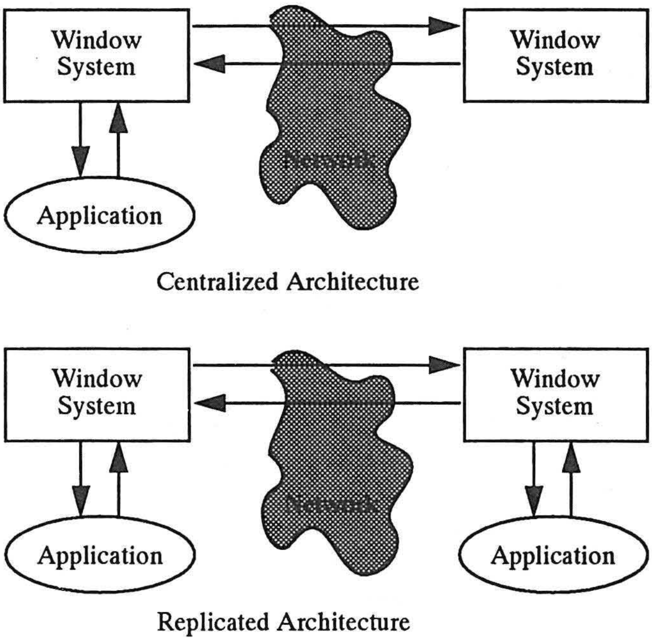

Figure 2: Shared Workspace Architectures

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.

7.2 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.

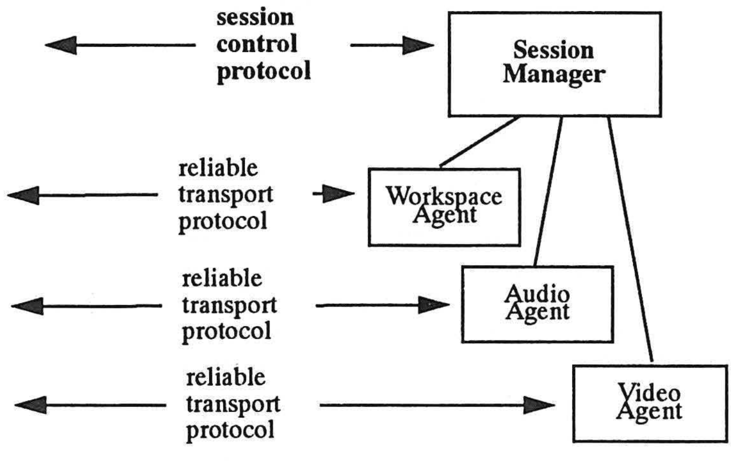

Figure 3: Session Control Architecture

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.

7.3 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.

7.4 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.

7.5 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?

8 Multimedia Interfaces

Stefan Noll, Fraunhofer-Gesellschaft, Darmstadt

8.1 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.

8.2 Position Statements

8.2.1 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.

8.2.2 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.

8.2.3 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

Dagstuhl-Seminar 9427: List of Participants

Sid **Ahja**
AT&T Bell Labs
Multimedia Research Dept.
Room 4F-601
101 Crawfords Corner RD
Holmdel NJ 07733
USA
sra@research.att.com
tel.: +1-908-949-55 69

Michael **Altenhofen**
Digital Equipment GmbH
CEC Karlsruhe
Vincenz-Prießnitz-Str. 1
D-76131 Karlsruhe
Germany
altenhof@nestvx.enet.dec.com
tel.: +49-721-690232

Gerold **Blakowski**
IBM ENC
Vangerowstr. 18
D-69115 Heidelberg
Germany
blak@dhdibmip.bitnet
tel.: +49-6221-59-44 20

Stephen L. **Casner**
University of Southern California
Information Sciences Institute
4676 Admiralty Way
Marina del Rey CA 90292
USA
casner@isi.edu
tel.: +1-310-822-15 11

Stefan **Covaci**
GMD-FOKUS
Hardenberg Paltz 2
D-10623 Berlin
Germany
covaci@fokus.berlin.gmd.dbp.de (X400)
tel.: +49-30-25499-270

Wolfgang **Effelsberg**
Universität Mannheim
Lehrstuhl Praktische Informatik IV
Seminargebäude A5
D-68131 Mannheim
Germany
effelsberg@pi4.informatik.uni-mannheim.de
tel.: +49-621-292-3131

José **Encarnação**
TH Darmstadt
Wilhelminenstr. 7
D-64283 Darmstadt
Germany
jle@idg.fhg.de
tel.: +49-6151-155-130

Steven K. **Feiner**
Columbia University
Department of Computer Science
450 Computer Science Building
500 W. 120th Street
New York NY 10027-7031
USA
feiner@cs.columbia.edu
tel.: +1-212 939-7083

James **Foley**
Georgia Institute of Technology
College of Computing
GVU Center
Atlanta GA 30332-0280
USA
foley@cs.gatech.edu
tel.: +1-404-853-06 71

Edward A. **Fox**
Virginia Polytechnic Institute
and State University
Computer Science Department
562 McBryde Hall
Blacksburg VA 24061-0106
USA
fox@fox.cs.vt.edu
tel.: +1-703 231-5113

Martin **Frühauf**
ZGDV - Darmstadt
Wilhelminenstr. 7
D-64283 Darmstadt
Germany
fruehauf@igd.fhg.de
tel.: +49 6151 155231

Arif **Gharfoor**
Purdue University
School of Electrical Engineering
West Lafayette IN 47907
USA
ghafoor@dynamo.ecn.purdue.edu
tel.: +1-315-494-06 38

Simon **Gibbs**
Université de Genève
Centre Universitaire d'Informatique
24 Rue du Général Dufour
CH-1211 Genève
Switzerland
simon@cui.unige.ch
tel.: +41-22-705-76 63

Bernd **Girod**
Universität Erlangen-Nürnberg
Lehrstuhl für Nachrichtentechnik
Causerstr. 7
D-91058 Erlangen
Germany
girod@nt.e-technik.uni-erlangen.de
tel.: +49-9131-85-71 00

Ulrich **Glowalla**
Universität Gießen
FB Informatik
Arndtstraße 2
35392 Gießen
Germany

Ralf G. **Herrtwich**
IBM ENC
Vangerowstr. 18
D-69115 Heidelberg
Germany
rgh@vnet.ibm.com
tel.: +49-6221-59-42 18

Thomas **Käppner**
IBM ENC
Tiergartenstr. 8
D-69020 Heidelberg
Germany
kaeppner@vnet.ibm.com
tel.: +49-6221-59-4424

Darin **Krasle**
Georgia Institute of Technology
College of Computing
GVU Center
Atlanta GA 30332-0280
USA
darin@cc.gatech.edu

Detlef **Kroemker**
IGD Darmstadt
Fraunhofer-Institut für
Graphische Datenverarbeitung
Wilhelminenstr. 7
D-64283 Darmstadt
Germany
kroemker@igd.fhg.de
tel.: +49-6151-155-1 32

Aurel A. **Lazar**
Columbia University
Department of Electrical Engineering
500 West 120th Street
New York NY 10027-6699
USA
aurel@ctr.columbia.edu
tel.: +1-212-854-1747

Thomas D. C. **Little**
Boston University
Dpt. of Elec. & Computer Engineering
44 Cummington St.
Boston MA 02115
USA
tdcl@bu.edu
tel.: +1-617-353-98 77

Blair **MacIntyre**
Columbia University
Department of Computer Science
450 Computer Science Building
500 W. 120th Street
New York NY 10027-7031
USA

Derek **McAuley**
University of Cambridge
Computer Laboratory
Corn Exchange Street
Cambridge CB2 3QG
Great Britain
dm@ct.cam.ac.uk
tel.: +44-223-33 44 16

Doris **Meschzan**
IBM ENC
Vangerowstr. 18
D-69115 Heidelberg
Germany
meschzan@vnet.ibm.com
tel.: +49-6221-59-45 49

Thomas **Meyer-Boudnik**
Universität Mannheim
Lehrstuhl Praktische Informatik IV
Seminargebäude A5
D-68131 Mannheim
Germany
mey@pi4.informatik.uni-mannheim.de

Klaus **Meyer-Wegener**
Technische Universität Dresden
Fakultät Informatik
Inst. SWT II
Mommsenstr. 13
D-01069 Dresden
Germany
kmw@freia.inf.tu-dresden.de
tel.: +49-351-45 75 - 3 83

Sape J. **Mullender**
Universiteit Twente
Faculteit der Informatica
Postbus 217
NL-7500 AE Enschede
The Netherlands
sape@cs.utwente.nl
tel.: +31-53 893 709

Mahesan **Nandikesan**
Columbia University
Department of Electrical Engineering
500 West 120th Street
New York NY 10027-6699
USA
mahesan@ctr.columbia.edu
tel.: +1-212-932-2458

A. Desai **Narasimhalu**
National University of Signapore
Institute of Systems Science
Heng Mui Keng Terrace
Singapore 0511
Singapore
desai@iss.nus.sg
tel.: +65 772-2002

Erich J. **Neuhold**
GMD-IPSI
Dolivostr. 15
D-64293 Darmstadt
Germany
neuhold@darmstadt.gmd.de
tel.: +49-6151-875 802

Stefan **Noll**
Fraunhofer-Institut für
Graphische Datenverarbeitung
Wilhelminenstr. 7
D-64283 Darmstadt
Germany
noll@igd.fhg.de

Maria José **Perez-Luque**
Boston University
Dpt. of Elec. & Computer Engineering
44 Cummington St.
Boston MA 02115
USA
mjpl@spiderman.bu.edu
tel.: +1-617-353-80 42

Stephen **Pink**
Swedish Institute of Computer Science
P.O. Box 1263
S-16428 Stockholm-Kista
Sweden
pink@sics.se
tel.: +46 8 752 15 59 / +46 8 38 60 15

P. Venkat **Rangan**
University of California at San Diego
Dept. of Computer Science and Engineering
Multimedia Laboratory
9500 Gilman Drive
La Jolla CA 92093-0114
USA
venkat@chinmaya.uscd.edu
tel.: +1-619 534-5419

Gerhard **Rossbach**
Springer Verlag
Tiergartenstr. 17
D-69121 Heidelberg
Germany
rossbach@spint.compuserve.com
tel.: +49-6221-487-652

Lawrence A. **Rowe**
University of California at Berkeley
Computer Science Division
Berkeley CA 94720
USA
larry@cs.berkeley.edu
tel.: +1-510-642-5117

Tom **Ruedebusch**
Universität Karlsruhe
Institut für Telematik
Kaiserstr. 12
D-76128 Karlsruhe
Germany
?@telematik.informatik.uni-karlsruhe.de
tel.: +49-721-608-47 90

Srihari **Sampathkumar**
University of California at San Diego
Dept. of CS &Engineering
Multimedia Laboratory
9500
La Jolla CA 92093-0114
USA
srihari@cs.ucsd.edu
tel.: +1-619 534-5419

Gerd **Schürmann**
GMD-FOKUS
Hardenbergplatz 2
D-10623 Berlin
Germany
schuermann@fokus.gmd.de
tel.: +49-30-25499-213

Eve M. **Schooler**
University of Southern California
Information Sciences Institute
4676 Admiralty Way
Marina del Rey CA 90292
USA
schooler@isi.edu
tel.: +1-310-822-15 11

Henning **Schulzrinne**
GMD-Fokus
Hardenbergplatz 2
D-10623 Berlin
Germany
hgs@fokus.gmd.de
tel.: +49 30 25499 219

Doug **Shepherd**
Lancaster University
Lancaster LA1 4YR
Gret Britain
doug@camp.lanes.ac.uk
tel.: +44-524-59-38 27

Ralf **Steinmetz**
IBM ENC
Vangerowstr. 18
D-69115 Heidelberg
Germany
steinmetz@dhdibmip.bitnet
tel.: +49-6221-59-42 80

Rüdiger **Strack**
Fraunhofer-Institut für
Graphische Datenverarbeitung
Wilhelminenstr. 7
D-64283 Darmstadt
Germany
strack@igd.fhg.de
tel.: +49-6151-155-224

Daniel C. **Swinehart**
Xerox PARC
3333 Coyote Hill Road
Palo Alto CA 94304
USA
swinehart.parc@parc.xerox.com
tel.: +1-415-812-4473

Fouad **Tobagi**
Stanford University
Department of Electrical Engineering
Stanford CA 94305
USA
tobagi@sierra.stanford.edu
tel.: +1-415-967-05 74

Zuletzt erschienene und geplante Titel:

F. Meyer a.d. Heide, H.J. Prömel, E. Upfal (editors):
Expander Graphs, Random Graphs and Their Application in Computer Science, Dagstuhl-Seminar-Report; 87; 11.04.-15.04.94 (9415)

J. van Leeuwen, K. Mehlhorn, T. Reps (editors):
Incremental Computation and Dynamic Algorithms, Dagstuhl-Seminar-Report; 88; 02.05.-06.05.94 (9418)

R. Giegerich, J. Hughes (editors):
Functional Programming in the Real World, Dagstuhl-Seminar-Report; 89; 16.05.-20.05.94 (9420)

H. Hagen, H. Müller, G.M. Nielson (editors):
Scientific Visualization , Dagstuhl-Seminar-Report; 90; 23.05.-27.05.94 (9421)

T. Dietterich, W. Maass, H.U. Simon, M. Warmuth (editors):
Theory and Praxis of Machine Learning, Dagstuhl-Seminar-Report; 91; 27.06.-01.07.94 (9426)

J. Encarnação, J. Foley, R.G. Herrtwich (editors):
Fundamentals and Perspectives of Multimedia Systems, Dagstuhl-Seminar-Report; 92; 04.07.-08.07.94 (9427)

W. Hoeppner, H. Horacek, J. Moore (editors):
Principles of Natural Language Generation, Dagstuhl-Seminar-Report; 93; 25.07.-29.07.94 (9430)

A. Lesgold, F. Schmalhofer (editors):
Expert- and Tutoring-Systems as Media for Embodying and Sharing Knowledge, Dagstuhl-Seminar-Report; 94; 01.08.-05.08.94 (9431)

H.-D. Ehrich, G. Engels, J. Paredaens, P. Wegner (editors):
Fundamentals of Object-Oriented Languages, Systems, and Methods, Dagstuhl-Seminar-Report; 95; 22.08.-26.08.94 (9434)

K. Birman, F. Cristian, F. Mattern, A. Schiper (editors):
Unifying Theory and Practice in Distributed Systems, Dagstuhl-Seminar-Report; 96; 05.09.-09.09.94 (9436)

R. Keil-Slawik, I. Wagner (editors):
Interdisciplinary Foundations of Systems Design and Evaluation, Dagstuhl-Seminar-Report; 97; 19.09.-23.09.94 (9438)

M. Broy, L. Lamport (editors):
Specification and Refinement of Reactive Systems - A Case Study, Dagstuhl-Seminar-Report; 98; 26.09.-30.09.94 (9439)

M. Jarke, P. Loucopoulos, J. Mylopoulos, A. Sutcliffe (editors):
System Requirements: Analysis, Management, and Exploitation, Dagstuhl-Seminar-Report; 99; 04.10.-07.10.94 (9440)

J. Buchmann, H. Niederreiter, A.M. Odlyzko, H.G. Zimmer (editors):
Algorithms and Number Theory, Dagstuhl-Seminar-Report; 100; 10.10.-14.10.94 (9441)

S. Heinrich, J. Traub, H. Wozniakowski (editors):
Algorithms and Complexity for Continuous Problems, Dagstuhl-Seminar-Report; 101; 17.10.-21.10.94 (9442)

H. Bunke, T. Kanade, H. Noltemeier (editors):
Environment Modelling and Motion Planning for Autonomous Robots, Dagstuhl-Seminar-Report; 102; 24.10.-28.10.94 (9443)

W. Maass, Ch. v.d. Malsburg, E. Sontag, I. Wegener (editors):
Neural Computing, Dagstuhl-Seminar-Report; 103; 07.11.-11.11.94 (9445)

G. Berry, W.P. de Roever, A. Poigné, A. Pnueli (editors):
Syncronous Languages, Dagstuhl-Seminar-Report; 104; 28.11.-02.12.94 (9448)

B. Becker, R. Bryant, O. Coudert, Ch. Meinel (editors):
Computer Aided Design and Test, Dagstuhl-Seminar-Report; 105; 13.02.-17.02.95 (9507)

D. Garlan, F. Paulisch, W. Tichy (editors):
Software Architectures, Dagstuhl-Seminar-Report; 106; 20.02.-24.02.95 (9508)

W.J. Cullyer, W.A. Halang, B. Krämer (editors):
High Integrity Programmable Electronic Systems, Dagstuhl-Seminar-Report; 107; 27.02.-03.03.95 (9509)

J. Gruska, T. Toffoli, H. Umeo, R. Vollmar (editors):
Cellular Automata, Dagstuhl-Seminar-Report; 108; 06.03.-10.03.95 (9510)

H. Alt, B. Chazelle, R. Seidel (editors):
Computational Geometry, Dagstuhl-Seminar-Report; 109; 13.03.-17.03.95 (9511)

W. Bibel, K. Furukawa, M. Stickel (editors):
Deduction, Dagstuhl-Seminar-Report; 110; 20.03.-24.03.95 (9512)

B. Freitag, C.B. Jones, Ch. Lengauer, H.-J. Scheck (editors):
Object-Orientation with Parallelism and Persistence, Dagstuhl-Seminar-Report; 111; 03.04.-07.04.95 (9514)

J. Doran, N. Gilbert, U. Mueller, K. Troitzsch (editors):
Social Science Microsimulation: A Challenge for Computer Science, Dagstuhl-Seminar-Report; 112; 01.05.-05.05.95 (9518)

R. Studer, M. Musen (editors):
Shareable and Reusable Problem Solving Methods, Dagstuhl-Seminar-Report; 113; 08.05.-12.05.95 (9519)

J. Blazewicz, K. Ecker, L. Welch (editors):
Scheduling in Computer & Manufacturing Systems, Dagstuhl-Seminar-Report; 114; 15.05.-19.05.95 (9520)

H. Beilner, G. Ciardo, C. Lindemann, K. Trivedi (editors):
Performance and Dependability Modeling with Stochastic Petri Nets, Dagstuhl-Seminar-Report; 115; 22.05.-26.05.95 (9521)

M. Aigner, J. Spencer, E. Triesch (editors):
Computing with Faulty Inputs, Dagstuhl-Seminar-Report; 116; 29.05.-02.06.95 (9522)

J.-R. Abrial, E. Börger, H. Langmaack (editors):
Methods for Semantics and Specification, Dagstuhl-Seminar-Report; 117; 05.06.-09.06.95 (9523)

W. Effelsberg, D. Ferrari, O. Spaniol, A. Danthine (editors):
Architecture and Protocols for High Performance Networks, Dagstuhl-Seminar-Report; 118; 19.06.-23.06.95 (9525)

Ph. Flajolet, R. Kemp, H. Prodinger, R. Sedgewick (editors):
Average-Case'-Analysis of Algorithms, Dagstuhl-Seminar-Report; 119; 03.07.-07.07.95 (9527)

D. Gustfield, T. Lengauer, C. Sander (editors):
Molecular Bioinformatics, Dagstuhl-Seminar-Report; 120; 10.07.-14.07.95 (9528)

J. Chomicki, G. Saake, C. Sernadas (editors):
Role of Logics in Information Systems, Dagstuhl-Seminar-Report; 121; 17.07.-21.07.95 (9529)

R.S. Boyer,A. Bundy, D.Kapur, Ch. Walther (editors):
Automation of Proof by Mathematical Induction, Dagstuhl-Seminar-Report; 122; 24.07.-28.07.95 (9530)