[Stei94f] *Ralf Steinmetz;* **The Next Generation of Distributed Multimedia System Architectures**; in "Multimedia"; Jose Encarnacao, Jim Foley (Hrsg.), Springer-Verlag, 1994, S. 173-186; Tagungsband zum Dagstuhl Seminar "Multimedia Systems and Applications", November 1992; Vorabdruck des Tagungsbands erschienen als ZGDV Bericht 71/93.

The Next Generation of Distributed Multimedia Systems

Ralf Steinmetz
IBM European Networking Center,Vangerowstr.18, 69115 Heidelberg, Germany
Fax: +49-6221-593400, e-mail: steinmet@dhdibmip.bitnet

Abstract

Distributed multimedia systems have been designed and implemented for several computer platforms, operating and window systems. All of them are conceived according to the paradigms of their specific environment. The Unix and the X window system with it's client(Xlib)-server approach is the most frequently used system for multimedia prototypes in the research community.

Interoperability between different systems and vendors are provided by means of common protocols and data (audio and video) coding formats. The next challenge is to conceive system structures for distinct environments which are nicely integrated with the various paradigms. This paper outlines such an approach which is currently under development at IBM ENC, Heidelberg. It provides distributed multimedia services on AIX, it enhances the OS/2 multimedia capabilities for distribution and integrates both as a distributed multimedia system.

Keywords

Multimedia, multimedia communication, distributed systems, distributed multimedia systems

1 Introduction: Environment

At the IBM European Networking Center (ENC) in Heidelberg, Germany, several HeiProjects have been established to develop prototypes that support distributed multimedia applications on RS/6000s under9 AIX as well as on PS/2s under OS/2 /11/. By "multimedia" we mean that continuous media such as audio and video is always taken into account /33; 33c/. Within this framework we encompass three related areas:

- The *HEIDELBERG CONTINUOUS-MEDIA REALM* (HeiCoRe) is concerned with providing local multimedia services to the applications. The essential services are the resource management and a real-time environment for stream handlers /14/. The resource management negotiates and guarantees the availability of the required resources such as memory, processing, bandwidth and delay. The real-time environment allows to fullfil these requirements by introducing real-time into conventional operating system environments /14/. In the initial phase we designed and implemented this system support for AIX as well as OS/2. Note, as soon as multimedia products such as IBM's Multimedia Presentation Manager/2 (MMPM/2) /17/ became available, we interfaced them replacing earlier prototype code.

- The *HEIDELBERG TRANSPORT SYSTEM* (HeiTS) transfers continuous media data between systems over today's networks such as Token Ring or FDDI in real time. The kind of media and it's properties are specified by the transport service user employing quality of service (QoS) parameters, which are then negotiated between the different HeiTS stacks. HeiTS comprises access to the communication adaptors, ST II as a network layer and HeiTP as a "thin" transport layer. HeiTS runs as a stream handler in the HeiCoRe environment and was the starting point of our integrated multimedia communication system's research and development.
- The *HEIDELBERG MULTIMEDIA APPLICATION TOOLKIT* (HeiMAT) interfaces to HeiCoRe, providing a uniform distribution mechanism on both platforms. It allows for abstractions of multimedia data, implements functions that are commonly needed in multimedia applications like synchronization and mixing of streams, and finally supplies the developer with these application-specific abstractions ready to use. It is aimed to supply AIX as well as OS/2 applications with the same homogeneous interface. This paper outlines the system architecture around HeiMAT which meets the requirements of these two distinct system environments. This is discussed in the framework of the respective evolutionary steps of distributed multimedia systems.

In the research community people sometimes tend to 'reinvent the wheel' as they ignore the evolving products related to their specific topic(s). Concerning audio and video in computing, there exist many 'local' multimedia products, for example, IBM's Multimedia Presentation Manager/2 /17/, Microsoft's Multimedia Extensions /22/ and Apple's Quicktime /4/. In several research driven projects similar capabilities were developed in conjunction with the specific application needs. Distributed multimedia systems should make use of these products by interfacing the available components.

Several experimental systems provide HeiMAT like functions. Among them are ACME /2/ VOX /1/, and Sventek's system /34/. Unlike HeiMAT, they are based solely on one environment which differentiates our system somewhat. One goal of the HeiProjects was to show this is not an insurmountable problem, rather a further abstraction from the system details.

Section 2 reviews the initial steps in distributed multimedia systems with a "hybrid" system structure, and Section 3 provides an overview of "unified" systems. In the next section the design demands as well as the available environments on two distinct platforms are outlined.

2 "Hybrid" System Structures

Early prototypes of distributed and local multimedia systems, such as the Integrated Media Architecture Laboratory (IMAL), conceived at Bell Communications Research in Red Bank /21/; or the Muse and Pygmalion system of MIT's Project Athena /16; 5; 26/, were based on a "hybrid" system structure /13/. In this framework, continuous media is mainly processed by devices located outside of the workstations. Most of the real-time processing is performed by dedicated processors and not by the main CPU(s). Traditionally, instead of sending video data to a workstation via a LAN for presentation in a window on the display, video data is sent over dedicated channels to a separate video monitor as shown in Figure 1. These devices are, however, attached to the workstation and controlled by the workstation's software.

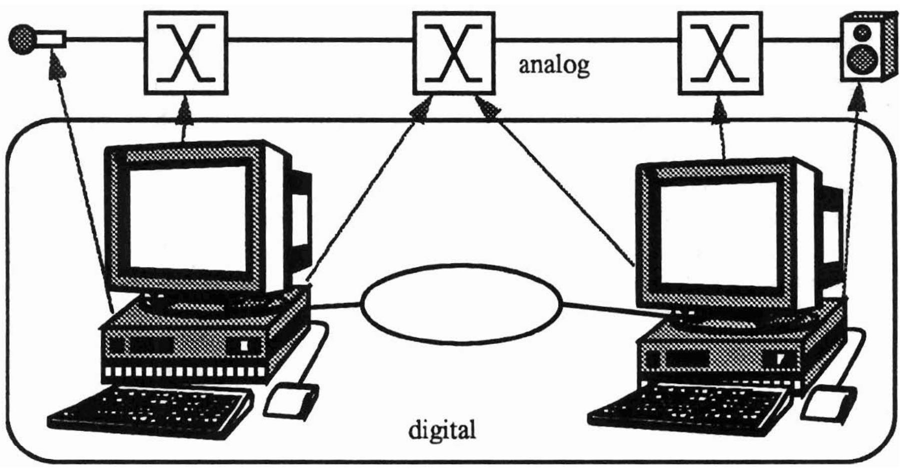

Figure 1: Hybrid System

The DiME (Distributed Multimedia Environment) project, carried out at the ENC in Heidelberg, was based on such a hybrid system structure. Continuous media is routed over dedicated channels using off-the-shelf hardware technology. Continuous and discrete media were integrated by connecting the audio and video equipment (e.g., CD players, VCRs) to a computer via an RS-232C interface. Devices could have been incorporated into the system as additional boards, for instance, the audio/video data was processed in the workstation using IBM's AVC and M-Motion adapters with respective system software /24; 23/. Further experiments included the ActionMedia 750 and ActionMedia II (DVI) technology /8/ for grabbing of images out of a video stream. DiME dealt with distributed, transparent access to multimedia resources like cameras and stored video sequences /27; 28; 29/. It aimed to provide an "easy, but rich" communication service as part of an application programming interface, manipulating data streams by controlling their sources and sinks in a heterogeneous computing environment. Synchronization has been a key issue in multimedia systems, it was also addressed in DiME by control of the devices located at the sources and sinks /31/.

Hybrid system structures require cost overheads for the additional devices not being part of the workstations, e.g., the required interconnections do not make use of available computer networks. The upgrade of a small system with about 10 involved workstations to a larger set-up with, e.g., more than 50 computers requires a considerable redesign of the hardware configuration, 'right-sizing' is difficult. In this hybrid approach the computer handles continuous media devices rather than the continuous media data. The audio and video data does not enter the computer after being generated; rather it passes through separate devices and it's own communication lines. Furthermore, continuous media data can not be manipulated with a fine granularity, operations like 'start' and 'stop' are supported.

This leads to some dedicated problems, for example, in synchronization. It is very difficult to achieve tight synchronization between discrete and continuous media. Discrete media and continuous media data are transmitted over different networks and processing nodes having different end-to-end delay characteristics. End-to-end delay over continuous media paths is typically shorter than for discrete media. By experiment, it is difficult and expensive (in terms of buffer capacity) to delay continuous media data delivered from devices like cameras or microphones. If discrete media is faster than continuous media, buffering and time stamping can be used to slow down the data stream, however these situations very rarely occur /32/.

3 "Unified" System Structures

The control of continuous media can be more immediate if all data passes through the computer system itself. This is only possible with digital data encoding. One of the first systems featuring digital audio in a computer environment was the Etherphone system developed at XEROX PARC /35/ in which an Ethernet is used for data communication and telephony. A similar early approach was used in a project by AT&T in Naperville /19; 20/ where a fast packet-switching network was directly attached to workstations. Enhancements to the UNIX operating system introduced the notion of "connectors" and "active devices" for handling continuous media.

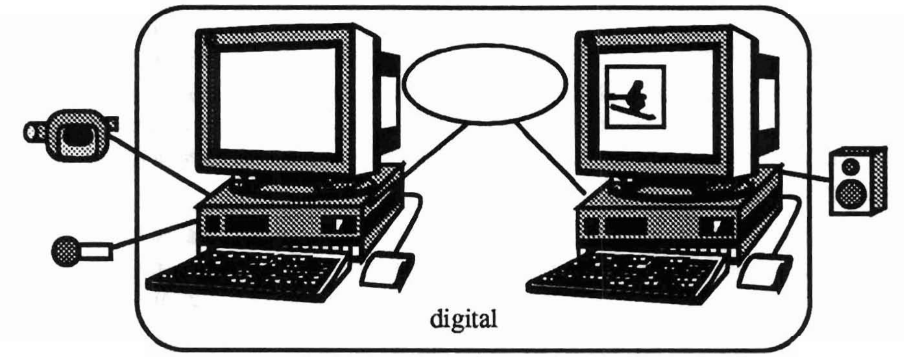

Figure 2: Unified System

Our follow-on project in it's first phase, the Heidelberg High-Speed Transport System (HeiTS), is based on this unified digital system structure /9/. Scheduling of continuous media data can either be done exclusively in software and/or by dedicated hardware, such as the ActionMedia II board (DVI). Both solutions require real-time techniques in a time-sharing environment (/10; 25/ similar to /3/).

Most of today's systems being conceived and implemented follow this approach. The design of the distributed systems follows the paradigms available in the respective environment. The communication between such systems is made possible by using the same protocols for continuous media, for example, ST II /12/ with HeiTP /7/, or a multimedia capable XTP /30/. The challenges of the next generation is to design and implement system structures which are appropriate for different computer architectures.

4 "Cross-Platform" System Structures

For the services provided by such an approach we foresee the following demands /15/:

- At the highest level it should be very *easy to develop applications*. The distributed system layer requires only the essential knowledge from the application in order to supply the demanded services. Then it takes care of all the details involved with establishment of data streams using devices in the distributed system.
- Coding technology is still evolving today, and it appears that committee stan-

dards compete with de-facto standards, where various implementations provide distinct algorithms for the same coding applications /33b/. The ISO JPEG standard defines compression and coding of single images like many available de-facto standards. The ISO MPEG video specification defines compression and to some extent competes with the CCITT H.261 and the DVI de-facto compression standard. MPEG-2 is still to be defined for video compression with higher quality. MPEG audio as well as the CCITT G.721 and 722 proposals are aimed at audio compression. Applications need to be developed without constraining to certain coding techniques by making use of the *presentation transparency* of the distributed system layer.

- Application must not see any difference in interfacing local or remote devices. By this *distribution transparency* the applications do not need to know explicitly the location of devices. Certainly by some applications it is required to be aware of the location and, e.g., to handle properly security aspects. Therefore, for the application it should be possible to make use of this knowledge but, only if it is required.

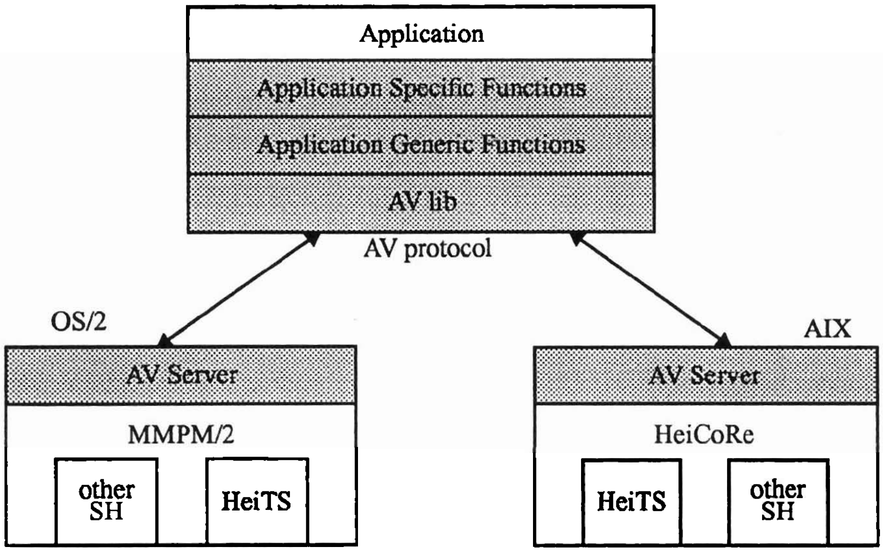

Figure 3: Cross-Platform Architecture

The *independence from multimedia devices* means to hide the characteristics of physical devices, allowing for the development of portable applications. For the remote control of a camera in order to change the position or to adjust the focus, most of the available camera control units require different interfaces. Some of them operate on a type of 'start and stop' semantics: the application is able to initiate or stop the movement. At other interfaces it is possible to specify the relative movement in term of, e.g., 'move north by 10 degrees'. Application programming should be independent of such implementation details. As a major interface metaphor, the application program interface should includes the notion of sources, sinks and streams.

This set of demands together with the distinct environments of the two different platforms impose hard design requirements on the system structure. The envisaged distributed system layer (HeiMAT) has to be a seamless integrator of the available paradigms on both platforms. The system structure is shown in the illustration above.

The OS/2 platform provides the MMPM/2 as the local multimedia extension to the operating system. MMPM/2 already allows the definition of sources and sinks of streams. HeiCoRe is used to add the resource management, it provides the reservation and scheduling of reserved resources in a distributed system taking into account the characteristics of the networks /36/. MMPM/2 closely interacts with the IBM's Presentation Manager which was designed for fast response times with a large set of functions in a local environment.

For the input and output of discrete media in a distributed UNIX environment, the X window system is the most widely used system. The layered approach of X is encompassing a client server approach. We envisage the multimedia support in a distributed environment to be architected in a similar manner to X. A continuous media server - which we call the AV server - communicates with the window system for presentation on the common display. The AV server encapsulates the functionality of HeiCoRe providing access to all types of stream-oriented multimedia devices/filters through a consistent interface. Native HeiCoRe applications coexist with the AV server and can interface to HeiCoRe. Similar to X, the communication between server and client is supported by an AV protocol which itself is hidden by an AVlib. A typical function set provided by the AVlib will include operations for creation, modification, connection, control, and destruction of the logical multimedia devices. These devices operate in a dedicated real-time environment. HeiMAT interfaces the AV server through the AVlib which can be seen as encapsulated into HeiMAT. Using this AVlib interface, HeiMAT will provide a higher-level API allowing for easy development of distributed multimedia applications. A prototype of the AV server and AVlib was developed in an object oriented C++ framework. This basic level of HeiMAT is, therefore, analogous to the X toolkit level.

However, HeiMAT will also support the development of multimedia interfaces through special widget sets similar to OSF/Motif. OS/2 already incorporates these through MMPM/2 /18/. These widgets provide the user of applications the same 'look and feel' for different multimedia applications. For this purpose in AIX, HeiMAT uses the X toolkit stack in conjunction with the AV server to provide it's services.

The Media Control Interface is the native OS/2 MMPM/2 component to be interfaced by applications. For the distribution of this interface an X-like server is being built around this native OS/2 Interface. It is known as the 'AV Server' which distributes the remote calls to the local multimedia devices via the Media Control Interface. Handles of the Presentation Manager's functions, which are used in the MMPM/2, can not be passed transparently as this window system has no client/server architecture such as X. Therefore the windowing application in OS/2 will always run at the client's station. In the first release of the distributed OS/2 system, HeiCam (Heidelberg Remote Camera Control) uses this transparent distributed access to remote multimedia devices.

As an alternative to our solution the distribution can be hidden under the Media Control Interface within the MMPM/2. There, at least two approaches can be followed:

- Each Media Driver establishes it's own communication with the remote entity(ies). As a consequence, there exist local as well as distributed Media Drivers. In our experiments it turned out that the commonalities of the various distributed Media Drivers are not taken into account. It must also be mentioned that no general scheme to name and to address the respective remote services exists.
- All kind of distribution is performed via the file system. In terms of OS/2 (unlike UNIX) all devices are not seen as being of similar input-output nature. The stream handlers are the sources and sinks of continuous data and, the file system is just one of these stream handlers. This approach is a suitable alternative for homogenous environments which focus on storage and retrieval applications. In such an approach the control of a remote cameras is handled similar to the access to files. At the programming interface we experienced that it is easier to operate on streams, sinks and sources than only on files. Therefore we did not follow this approach.

Due to the mentioned reasons we decide to build a server around the local multimedia functions and did not add distribution capabilities to individual entities within this multimedia extension.

In our Heidelberg multimedia system, the HeiRAT component /36/ is in charge of the resource management. HeiRAT accepts quality of service (QoS) requests from the AVlib (as part of HeiMAT). HeiRAT can be seen as part of the AVserver which serves for the QoS demands as interface to the whole distributed system. It makes use of the ST II flow specification to negotiate them among the whole set of involved system components /7b/. It provides a QoS calculation by optimizing one QoS parameter dependent on the resource characteristics. Subsequently resources are reserved according to the QoS guarantees. At the actual data transfer phase resources are scheduled (in the real-time environment) to provide these guarantees.

As a part of the HeiProjects we architected this system structure and defined HeiMAT to be the encapsulation of the client multimedia library. However, HeiMAT goes beyond the existing multimedia services of MMPM/2 and the related AVlib interfaces in providing abstractions such as a video conferencing module that serves as a building block for multimedia applications.

5 Conclusion

Distributed multimedia system structures evolved from *hybrid* to *unified* approaches. In several European projects around ESPRIT, RACE and DELTA as well as in national initiatives like BERKOM /37/, systems are implemented which interconnect UNIX and UNIX-like environments of different vendors. The next generation will comprise *cross-platform* solutions which additionally cover different system environments like AIX and OS/2.

This paper discusses an approach of such a cross-platform solution. Implementation of different basic components as well as the design of upper layer toolkit like modules is in progress. A first system, together with a conferencing application was shown at COMDEX, Las Vegas, November 1992 and at CeBIT, March 1993. The current version (demonstrated at CeBIT'93) includes the main ideas as outlined in this paper.

As a matter of fact we were forced to provide the interoperability of the different platforms, therefore, in order to avoid a duplication of work we developed this concept of portability and interoperability between various platforms. We experienced that the concept of the AVlib and AVserver is excellent to meet the requirements of ease of development, distribution transparency, presentation transparency and independence from multimedia devices. We see no reason why our system can not be ported to non-IBM platforms.

We are still in the phase of redesign of the interface to the application including the Application Specific and Generic Functions. We have two different environments and implementations which we are still not satisfied with. Each approach tends to either be very effective and closely related to one specific operating system environment or it tends to be too generic for being efficient to implement a whole set of different applications on various machines. It is still an open issue how such an excellent interface in terms of, e.g., an object oriented class hierarchy should look like in order to support the set of interactive as well as retrieval like multimedia applications.

I would like to thank all the application and applications support teams which are in charge of the distributed system in the HeiProjects. Thomas Käppner provided his in-depth X and UNIX experience in designing a toolkit approach for HeiMAT. Jürgen Falter and Peter Sander contributed with their MMPM/2 experience. Thomas Meyer reminded us of all possible obstacles and helped with his exceptional multimedia synchronization and abstractions experience. Ralf Guido Herrtwich intensively pushed all the initial steps and discussed in detail many issues of system structure. Dietmar Hehmann made extensive use of his profound OS/2 as well as AIX background in order to get an aligned solution for both systems. Ian Marsh provided substantial contributions to the whole paper in it's final version.

6 References

/1/ B. Arons, C. Binding, K. Lantz, C. Schmandt. The VOX Audio Server. 2nd IEEE COMSOC International Multimedia Communications Workshop, Montebello, Quebec, Canada, Apr. 1989.

/2/ David Anderson, Pamela Chan; Toolkit Support for Multiuser Audio/Video Applications; 2nd International Workshop on Network and Operating System Support for Digital Audio and Video, Heidelberg, November 18-19, 1991.

/3/ David P. Anderson; Meta-Scheduling for Distributed Continuous Media; Computer Science Division (EECS) Report No. UCB/CSD 90/599, U. C. Berkeley, Berkeley CA, October 1990.

/4/ Apple; QuickTime Developer's Kit Version 1.0; Apple Document Number 030-1899.

/5/ George Champine, Daniel Geer, William Ruh; Project Athena as a Distributed Computer System; IEEE Computer, vol.23 no.9, September 1990, pp.40-51.

/6/ Andreas Cramer, Manny Farber, Brain McKellar Ralf Steinmetz; Experiences with the Heidelberg Multimedia Communication System: Multicast, Rate Enforcement and Performance; IFIP European High Performance Networking Workshop (ehpn'92), Liege, Belgium, December 1992.

/7/ Luca Delgrossi, Christian Halsstrick, Ralf Guido Herrtwich, Heiner Stuettgen; HeiTP - A Transport Protocol for ST-II; IBM ENC Technical Report, 1992.

/7b/ Luca Delgrossi, Ralf Guido Herrtwich, Frank Oliver Hoffmann: An Implementation of ST-II for the Heidelberg Transport System, IBM Technical Report 43.9303, IBM European Networking Center, Heidelberg, 1993.

/8/ Kevin Harney, Mike Keith, Gary Lavelle, Lawrence D. Ryan, Daniel J. Stark; The i750 Video processor: A Total Multimedia Solution; Communications of the ACM, vol.34, no.4, April 1991, pp.64-78.

/9/ Dietmar Hehmann, Ralf Guido Herrtwich, Werner Schulz, Thomas Schuett, Ralf Steinmetz; HeiTS - Architecture and Implementation of the Heidelberg High-Speed Transport System; 2nd International Workshop on Network and Operating System Support for Digital Audio and Video, Heidelberg, November 18-19, 1991.

/10/ Ralf Guido Herrtwich; The Role of Performance, Scheduling, and Resource Reservation in Multimedia Systems; Proc. Operating Systems in the Nineties and Beyond, Lecture Notes of Computer Science, Springer, 1991.

/11/ Ralf Guido Herrtwich; The HeiProjects: Support for Distributed Multimedia Applications; IBM Technical Report, no.43.9206, March 1992.

/12/ Ralf Guido Herrtwich; The Evolution of ST II; Position Paper for the Dagstuhl Seminar, Novembner 1992.

/13/ Lutz Henkel, Heinrich J. Stuettgen; Transportdienste in Breitbandnetzen; GI Conference, Communication in Distributed Systems, Mannheim, Germany, E. Effelsberg, H.W. Meuer, G. Mueller (Ed), Springer Verlag, pp.96-111, February, 1991.

/14/ Ralf Guido Herrtwich, Lars Wolf; A System Software Structure for Distributed Multimedia Systems; 5th Acontinuous media SIGOPS European Workshop, Le Mont Saint-Michel, France, September 1992.

/15/ Dietmar Hehmann, Thomas Kaeppner, Ralf Steinmetz; An Introduction to HeiMAT: The Heidelberg Multimedia Application Toolkit 3rd International Workshop on Network and Operating System Support for Digital Audio and Video, San Diego, CA, Nov. , 1992.

/16/ Matthew E. Hodges, Russel M. Susnett, Mark S. Ackerman; A Construction Set for Multimedia Applications; IEEE Software Magazine, January 1989, pp. 37-43.

/17/ IBM; Multimedia Presentation Manager/2: Programming Reference; IBM Document From Number 41G2920, 1992.

/18/ Thomas Käppner; Personal Communication; August 1992.

/19/ Wu-Hon F. Leung, Gottfried W. R. Luderer; The Network Operating System Concept for Future Services; AT & T Technical Journal, vol.68, no.2, April 1989, pp. 23-35.

/20/ W.H. Leung, T.J. Baumgartner, Y.H. Hwang, M.J. Morgan, S. C. Tu. A Software Architecture for Workstation Supporting Multimedia Conferencing in Packet Switching Networks. IEEE Journal on Selected Areas in Communication, vol.8, no.3, April 1990, pp. 380-390.

/21/ L.F.Ludwig, D.F.Dunn; Laboratory for Emulation and Study of Integrated and Coordinated Media Communication; Frontiers in Computer Technology, Proc. of the ACM SIGCOMM '87 Workshop, August 11-13,1987.

/22/ Microsoft Corporation; Microsoft Windows Multimedia Programmer's Reference; Microsoft Press, 1991.

/23/ M-Motion Video Adapter/A, User's Guide, Product Description; IBM 1990.

/24/ Daniel J. Moore; Multimedia Presentation Development using the Audio Visual Connection; IBM Systems Journal, Vol.29, No.4, 1990, pp.494-508.

/25/ Andreas Mauthe, Werner Schulz, Ralf Steinmetz; Inside the Heidelberg Multimedia Operating System Support: Real-Time Processing of Continuous Media in OS/2; IBM ENC Technical Report, October 1992.

/26/ W.E. Mackay, W. Treese, D. Applebaum, B. Gardner, B. Michon, E. Schlusselberg, M. Ackermann, D. Davis Pygmalion: An Experiment in Multimedia Communication Proceedings of SIGGRAPH 89, Boston, July 1989.

/27/ Johannes Rueckert, Hermann Schmutz, Bernd Schoener, Ralf Steinmetz; A Distributed Multimedia Environment for Advanced CSCW Applications; IEEE Multimedia '90, Bordeaux, November 15-17, 1990.

/28/ Ralf Steinmetz, Reinhard Heite, Johannes Rueckert, Bernd Schoener; Compound Multimedia Objects - Integration into Network and Operating Systems; International Workshop on Network and Operating System Support for Digital Audio and Video, International Computer Science Institute, Berkeley, November 8-9, 1990.

/29/ Bernd Schoener, Johannes Rueckert, Ralf Steinmetz; Media Related Requirements for Communication Services; 6th IEEE International Workshop on Telematics, Corea, September 1991.

/30/ Jochen Sandvoss, Thomas Schuett, Markus Steffen, Ralf Steinmetz; XTP und Multimedia?; GI Conference, Communication in Distributed Systems, Muenchen 1993.

/31/ Ralf Steinmetz; Synchronization Properties in Multimedia Systems; IEEE Journal on Selected Areas in Communication, vol. 8, no. 3, April 1990, pp. 401-412.

/32/ Ralf Steinmetz, Multimedia Synchronization Techniques: Experiences Based on Different System Structures; IEEE Multimedia Workshop '92, Monterey, CA, USA, April'92.

/33/ Ralf Steinmetz, Ralf Guido Herrtwich; Integrierte verteilte Multimedia-Systeme; Informatik Spektrum, Springer Verlag, vol.14, no.5, October 1991, pp.280-282.

/33b/ Ralf Steinnmetz; Compression Techniques in Multimedia Systems; IBM ENC Technical Report no.43.9305, 1993; and accepted for publication in acm/Springer Multimedia Systems Journal, 1994.

/33c/ Ralf Steinmetz; Multimedia-Technology: Introduction and Fundamentals (book, in German); Springer-Verlag, October 1993.

/34/ J.S.Sventek; An Architecture for Supporting Multimedia Integration; IEEE Computer Society Office Automatition Symposium, April 1987, pp.46-56.

/35/ Daniel C. Swinehart; Telephone Management in the Etherphone System; IEEE Globecom'87, 1987, pp. 30.3.1-30.3.5.

/36/ Carsten Vogt, Ralf Guido Herrtwich, Ramesh Nagarajan; HeiRAT: The Heidelberg Resource Administration Technique, Design Philosophy and Goals; IBM ENC Technical Report no.43.9101, 1991.

/37/ BERKOM Breitband Kommunuikation im Glasfasernetz; H.Ricke, J.Kanzow (Hsrg.), R. v. Decker's Verlag, 1991.