Johannes Rückert, Bernd Schöner, Ralf Steinmetz, Hermann Schmutz:
[RSSS90] *A Distributed Multimedia Environment for Advanced CSCW Applications*. In: *IEEE Multimedia '90, Bordeaux, France*, November 1990.

A Distributed Multimedia Environment for Advanced CSCW Applications

Johannes Rückert, Hermann Schmutz, Bernd Schöner, Ralf Steinmetz

IBM European Networking Center, Tiergartenstr. 8, 6900 Heidelberg,
P.O.Box 10 30 68, tel: +49 6221 404280, FAX: +49 6221 404450
West-Germany, e-mail: RUECKERT at DHDIBM1

Scenario

The most sophisticated and advanced type of applications in a distributed multimedia environment can be found within the wide spectrum of computer supported cooperative work (CSCW) [3; 14]. Areas of CSCW cover interactive tutoring, integrated surveillance systems, conferencing systems and joint working sessions.

As an example, let us consider a scenario where two or more journalists write an article for a newspaper. A reporter may be at some place around the world collecting local information about an important event. Another journalist is sitting in his office at the headquarters while the first one starts typing the article incorporating some local information provided by pictures, audio and video data. The partner in the headquarters may contribute with data form databases and archives. Both journalists see the same document at the same time and may also talk to each other over the network using microphones and loudspeakers. They can discuss their cooperative work like sitting together in one room and at the end the article is immediately available at the headquarters as an integrated multimedia document.

CSCW Demands - DiME Objectives

From this scenario we can derive most of the relevant requirements for our Distributed Multimedia Environment project (DiME). A first important point is the existence of a *high-speed network* providing real-time capabilities for the various media streams. The real-time transmission of huge amounts of data is essential for cooperative work. Another issue is the *support of heterogeneity* in different areas. This includes hardware architectures, operating systems, communication facilities, devices and coding schemes. Whereas the journalist in the headquarters may use HDTV, HIFI audio, a powerful workstation and a mainframe, the correspondent outside may operate with a S-VHS camera and low quality audio equipment attached to a portable. The concept of *transparency* in distributed systems covers several areas and denotes some degree of independence of inherent properties in distributed environments. In the context of our project, we are basically interested in distribution transparency, access transparency, name transparency, location transparency, performance transparency, and presentation transparency.

In the distributed multimedia system, each node is an autonomous system. The cooperation between the nodes is based on the principle of transparent resource sharing with the cooperation participants. Each node has complete control over the local environment and determines the degree of cooperation with other partners. It also controls the granularity of access to the local resources by other partners. With this background we can conclude that access to shared resources must be protected against unauthorized usage. The sketched system environment so far comes from the idea of supporting multimedia within existing operating systems. The users must be able to run "old" existing applications as well as "new" multimedia applications. The new multimedia environment must coexist with the existing systems while the bandwidth of perception is enhanced by adequate presentation of all kind of media at the user interface level. Relations like *synchronization* in time and location must be established between different multimedia entities and a "relatively easy" *application program interface* for the design of distributed multimedia applications must be provided.

DiME Approach

One problem in designing a "good" distributed multimedia system is to have and get experience as user, as programmer and as system architect in this environment. Experience can only be gained with a real distributed multimedia system or a prototype. Unfortunately such systems do not exist today, but prototypes can be built on top of existing hardware and operating systems.
We gained experience in the area of distributed systems with DACNOS [5], and in the area of local multimedia systems by integration of plug-in boards [1; 10] and external devices like computer controllable VCR, optical memory and cameras and by designing and implementing local control applications for those devices. First fundamental design issues were already published in [13; 12]. Our approach to proceed has been to design and implement the prototype step by step, gaining experience with the current system day by day, and use the experience for the progress of the design for an architected system.

High-speed networking is up to now a challenging research area, but no integrated network with the appropriate protocols for point-to-point and multicast transmission of high-quality video together with audio and data is available today. In the past a heterogeneous network, consisting of digital and analog transmission media was used for early experiments. Transition to a fully digital network is on the way. The design enables the usage of networks like FDDI or B-ISDN, a channel management is conceived to make us independent from the basic communications hardware. A layered systems structure is used, to *hide the heterogeneity* of a specific level from the higher levels. For example the intended channel management hides heterogeneous data transmission facilities of different vendors from the upper layers. *Autonomy* will be provided, in giving the complete control of the local equipment and system software to the local user by the network operating system approach. Remote control is granted by capabilities which can be withdrawn by the owner of the device at any time. *Coexistence* of the multimedia systems with existing applications is guaranteed, since the multimedia system is designed as an add-on to the existing system. The different aspects of *transparency* especially the problem of distribution transparency and its related problems pose high demands to the system structure. The Remote Service Call (RSC) provides a unique interface for service calls in a heterogeneous environment in a distribution transparent form [4] but other communication services like TCP/IP are also investigated for this purpose. DiME is conceived for high *performance* local multimedia processing, storage and presentation since dedicated devices are used and integrated. The performance in the distributed case depends heavily on the isochronous data transfer characteristics of the interconnected high-speed network(s). To design an *application programming interface* we chose an class-based approach which may be extended towards object-orientation, which enables the handling of various multimedia objects and devices [6]. The composition of simple multimedia objects like joint windows, cameras, microphones, voice, video and data connections to a composed conferencing object requires means for object management and synchronization. The concept for *synchronization* of multimedia data proposed in [12] was adopted for the project.

DiME Architecture and Components

Building up a multimedia lab requires a multimedia workstation being today dedicated multimedia equipment which can be controlled by a conventional workstation. For our lab, we currently use PS/2 systems as basic workstations, running OS/2. Figure 1 shows the overall architecture of the DiME system.

Special internal and external hardware for input, output, storage and processing of video and audio data is attached to the PS/2. The *external attachments* like VCR, CD-player, optical memory, camera, monitor, microphone and loudspeaker can be controlled by the computer via a conventional RS-232C interface. *Internal equipment* for full-motion video display, frame grabbing, audio storage, processing and presentation is available as plug-in boards [1; 10] controlled by device drivers.

All these devices are represented by *device servers* within DiME. The device servers are processes running under the operating system and controlling directly the various multimedia devices. They are the representatives for the multimedia hardware, offering unique control functions to higher layers of the DiME system. For example, all storage device servers (VCR, OM, CD-player, M-Motion, AVC) support the same basic functions play, record, stop, pause, goto and others. Similar functions are available for input-, output- and processing devices. The device servers are designed to hide the differences of the control interfaces of the underlying devices, to meet our stated transparency concerning device access.

Besides device control, *data communication* is needed to interconnect the devices as multimedia data sources and sinks. Since transmission of high speed data with real-time constraints is required, but up to now digital networks for these purposes are not available, we made first experiments with a mixed network approach. Data sources can be connected to sinks and depending on the type of data and their encoding (digital or analog), data is transported via dedicated wire connections or via the processor bus. For example, if the M-Motion board is used to display an image from the harddisk onto the screen, the digital image is moved from the harddisk controller to the M-Motion board. If the video from the external camera is to be displayed on an external monitor, a connection is setup. For this purpose, an external local analog crossbar switch not shown in figure 1 controlled by the computer is used. Some internal devices do have analog input connectors as well, so they are connected to the local crossbar switch.

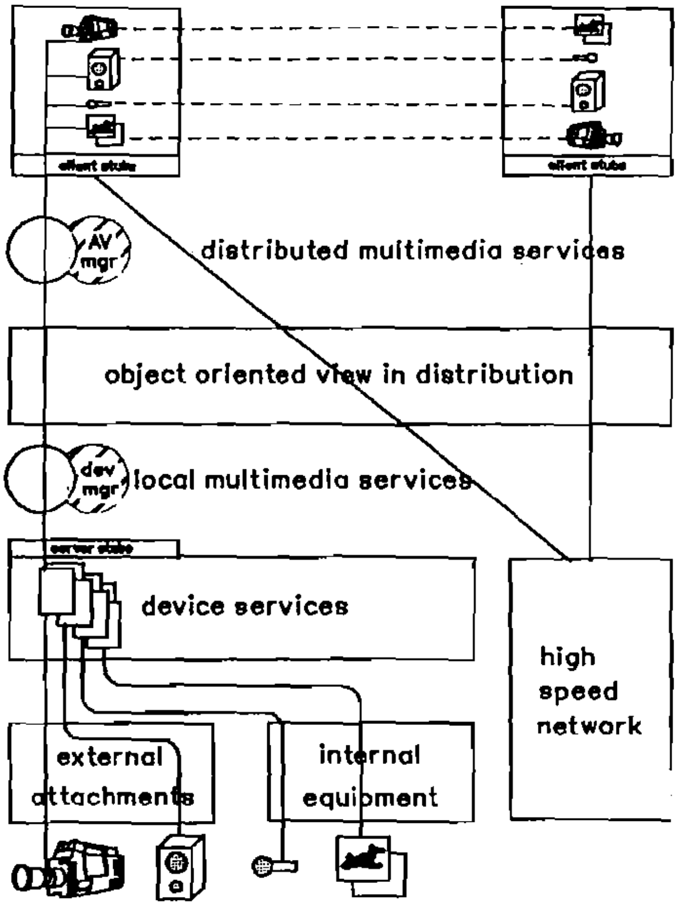

Figure 1. Overall structure of DiME

The *device manager* is designed to configure the local DiME system components, and to control the interconnections between them, especially the device servers.

With this, we have a local multimedia system, which is the basis to allow for distribution, i.e. control and usage of remote devices and services, receive and transmit data to/from remote workstations. Since the DiME architecture allows the usage of different networks, their specific peculiarities have to be "hidden" from the user, i.e. made transparent for usage. The user selects a device as multimedia source and another as a sink. He therefore sends inquiries to the device manager at the involved nodes to get the access rights for these devices from the DiME system. Then he connects data in- and output using the connect command, which is performed by the A/V channel manager for audio/video connections. The channel manager looks up the device's description data, decides for the validity of the connection, and selects the connection path or invokes the involved data transport system. If in our first prototype both devices use analog data and an analog connection between them is available, the appropriate calls to the controllers of the crossbar switches are made (the local, remote and intermediate switch). If the devices require digital connections, the network system is used to establish a connection of appropriate quality and data conversion between source and destination. If analog to digital or vice versa conversion is needed the AV manager checks for a suited conversion device within the network and sets up the data paths. Otherwise the connection call fails.

The user of the DiME system becomes independent from the underlying transport system for analog or digital data and the way multimedia data is presented within DiME. Transport services are accessed in a unique way, differing perhaps in their quality of service. If, for example a full motion video is established, but only a slow LAN is available a slow-motion transmission may result.

To enable the user to access devices and data in a more appropriate way, i.e. to think of audio and video as a music play or piece of film, an *object oriented view in distribution* is supported by the DiME system [6]. The user is no more forced to inquire for multimedia devices and their content, but can directly access multimedia data via a name, once it is encapsulated as a multimedia object [11]. Also compositions of such objects are supported, to enable easy handling of more complex data. E.g. a multimedia object called "information" could consist of a video sequence together with some textual and verbal explanation at the same time. Devices themselves can also be treated as such objects, to enable their composition for complex tasks like conferencing.

In figure 1 an example for two conference devices, consisting of audio/video in- and output are shown. The devices reside at different physical nodes. Each conference device object controls its local parts (camera, video window, microphone, loudspeaker). Data connections are set up by the AV manager like described above. The composed conference devices are controlled by the applications, which also can exchange their access rights to them.

DiME Prospect

At the time being we assembled one "high end" workstation with controls many external devices (VCR, CD-player, OM) and the AVC board. Two "low end" work stations are equipped with M-Motion boards for video-in-window presentation. Up to now the digital audio storage processing and storage capabilities can not be used, since the interface for OS/2 Presentation Manager is not completely implemented. One crossbar switch handles analog audio/video connections, digital interconnection is provided via a token ring using RSC. Control applications for all external devices running under the Presentation Manager in OS/2 were implemented and tested, and first experience could be gained. Since communication between all processes is done via RSC, applications run at remote PS/2 to control the devices of any involved the workstations. A control interface for a CD audio player has been ported to X-windows running within AIX as a first distributed heterogeneous experiment. Object orientation is still to be realized, as well as a unique communication management (A/V manager), but distributed demonstrations are running.

The design and incorporation of multimedia dependent features within the ENC's high speed network is one of the next steps in order to realize a fully digital solution. Related projects like [9; 7; 8; 2] and especially IMAL at Bellcore, Red Banks, have demonstrated the importance of distributed multimedia systems. So, what is "new" with/in DiME? DiME is conceived as an distributed multimedia add-on to existing systems, it is based on the network operating system approach. To the knowledge of the authors it is the first approach to incorporate heterogeneity at many levels in the distributed multimedia environments covering different native operating systems.

References

[1] *IBM Audio Visual Connection, Product Documentation;* IBM 1990.

[2] Frank Ball, David Hutchinson, Andrew Scott, Doug Shepard; *A Multimedia Network Interface;* 3rd IEEE COMSOC International Multimedia Workshop (Multimedia '90), Bordeaux, France, November 24-37, 1989.

[3] *Proceedings of Computer Supported Cooperative Work 1988.* Portland, Oregon, September 26-28, 1988.

[4] H.Eberle, K.Geihs, A.Schill, H.Schmutz, B.Schöner; *Generic Support for Distributed Processing in Heterogeneous Networks;* in Hector, Heterogeneous Computers Together, Volume II: Basic Projects, Springer-Verlag, 1988.

[5] Kurt Geihs, Ulf Hollberg; *Retrospective on DACNOS;* Communications of the acm, vol. 33, no. 4, pp. 439-448, 1990.

[6] Ulf Hollberg, Herbert Eberle, Kurt Geihs, Reinhard Heite, Hermann Schmutz; *An Object Oriented View Of Distribution;* Technical Report No. 43.9004, IBM European Networking Center, 1990.

[7] Andy Hopper; *Pandora - an experimental system for multimedia applications;* acm Operating System Review, April 1990, pp.19-34.

[8] Matthew E. Hodges, Russel M. Susnett, Mark S. Ackerman; *A Construction Set for Multimedia Applications;* IEEE Software Magazine, Jan. 1989, pp. 37-43.

[9] 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.* 2nd IEEE COMSOC International Multimedia Communications Workshop, Montebello, Quebec, Canada, Apr. 1989.

[10] *M-Motion Video Adapter / A, User's Guide, Product description;* IBM 1990.

[11] Johannes Rückert, Bernd Schöner, Ralf Steinmetz; *Compound Multimedia objects - Integration into network and Operating Systems;* Int. Workshop on Network and Operating System Support for Digital Audio and Video, International Computer Science Institute (ICSI), berkley, CA, Nov. 8-9, 1990.

[12] Ralf Steinmetz; *Synchronization Properties in Multimedia Systems;* IEEE Journal on Selected Areas in Communication, vol.8, no.4, April 1990.

[13] Ralf Steinmetz, Hermann Schmutz, Bernd Schöner, Michael Wasmund; *Generic Support for Distributed Multimedia Applications;* IEEE ICC 90, Atlanta, April 1990.

[14] Paul Wilson; *Key Research in Computer Supported Cooperative Works (CSCW);* in Reserach into Networks and Distributed Applications, R.Speth (editor), Elsevier Science Publishers B.V., North-Holland, 1988.