Audio and Video in Distributed Computer Systems: Why and How?

Ralf Guido Herrtwich

TR-90-026

July 2, 1990

Abstract

Technological advances allow computer systems to handle "continuous media" such as audio and video in addition to "discrete media" such as text and graphics. As with the introduction of computer graphics ten years ago, the integration of continuous media will extend the range of computer applications and change existing paradigms for computer usage and programming. Distributed computer systems that are capable of handling continuous media can (1) unify the methods of information distribution, (2) personalize information services through interactive access and individual information selection, and (3) make information presentation more effective. The major obstacles to using continuous media in today's computer systems are performance limitations. In addition to high-capacity and high-speed hardware, system software is needed that meets the real-time demands of audio and video, and that provides application interfaces which take the special requirements of these new data types into account.

1. INTRODUCTION

Ever since computer pioneers found it too cumbersome to communicate with their machines through binary code, text has been the primary means of man-machine interaction. I/O equipment, from card readers in the 50's to teletypes in the 60's to CRT terminals in the 70's, facilitated the exchange of data between computers and their human users by means of characters. "Media" is the term used to denote systems of communication between humans and computers – and among humans using computers as communication tools. *Text* was the only medium available during the first three decades of computing. Not until the 1980's have new media – *graphics* and *still images* – made their entrance in the computing arena, again facilitated by new I/O devices – bitmap displays and pointing devices – available to a large user community. Now, in the 1990's, the advent of digital signal processors, together with the ability to store, exchange, and handle large data quantities in today's computer systems, enables the introduction of yet two other media: sound (*audio*) and motion picture (*video*).

The fairly recent integration of graphics into computing makes the importance and impact of new media evident: media determine *how* and *for what purpose* computers are used. Graphics made new applications such as drawing and geometric modeling possible. Computers took over the function of traditional design tools, improving quality by integrating accuracy and feasibility checking, reducing human workload through automation, and extending the capabilities of existing tools by applying the versatility of the computer to the new application domain. Beyond this, graphics changed the method of interaction with the computer. Icons, windows, and the desktop paradigm all resulted from the new media, and changed the way man and machine could communicate.

In this short paper, we elaborate on why we consider it desirable to have audio and video available in tomorrow's computer systems and on how such an integration should be achieved. Section 2 deals with the potential advantages and implications of having audio and video available as a means for man-machine interaction. In Section 3 we describe the key issues in realizing systems that support these media. The concluding Section 4 summarizes our own work towards the integration of audio and video.

2. ADVANTAGES OF AUDIO AND VIDEO INTEGRATION

Integrating audio and video in today's distributed computer systems obscures the lines between three areas – *computing*, *telecommunications*, and *mass media* – which constitute major pillars of the post-industrial society. The social and economic consequences of this development will be profound. If such changes were purely technology-driven, they would be highly questionable. The information market, however, because of the flexibility of its products, is perhaps more driven by consumer demands than any other. Therefore, such an upheaval – even greater than that seen with the introduction of VCRs, compact discs, and electronic mail services – will only occur if consumers see advantages in the integration. Such advantages (which usually go hand in hand) can be

- the reduction of costs for existing services,
- the improvement of the quality of existing services,
- the possible interaction between existing services, or
- the availability of new services.

With the availability of audio and video in distributed computer systems, potential advantages result from new possibilities for information distribution, selection, presentation, and processing.

2.1. Unified Information Distribution

The telecommunications industry worldwide is moving towards the establishment of *integrated services digital networks* (ISDNs) [1], realizing that once all data has a common digital representation there is no need to maintain the existing multitude of channels for information distribution. With the emerging optical *broadband ISDN* (B-ISDN) technology [2-4], such a multi-purpose network can carry not only text, data, and voice, but also video and high-quality audio, making it possible to offer television and radio services as well. In the U.S. and Japan, where telecommunications and cable TV companies operate separately, the introduction of B-ISDNs implies heavy competition. In Europe, where the different services are usually provided by the same state-controlled organization, the transition will be much smoother.

The high bandwidth and pervasiveness of future B-ISDNs make it possible for them not only to integrate existing communication networks, but also to replace traditional non-electronic channels of information distribution. For example, instead of renting videos from a local video store,

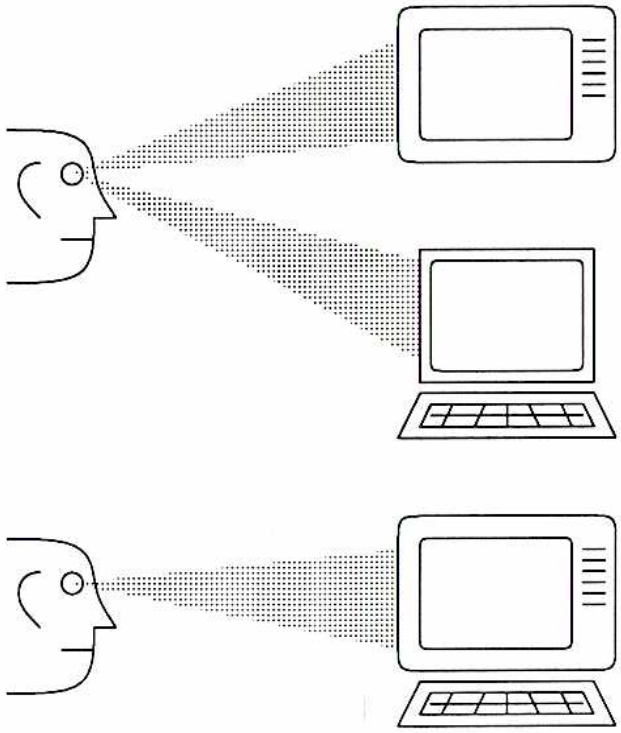

Figure 1: Information distribution.
Above: Traditional separation of devices and services. Below: Unification.

users can download them from a remote file server. This increases the availability of a particular video program (the cassette may have been rented by somebody else, but the master video on the file server is always accessible) and reduces the access time, *i.e.*, the use of the network improves the overall service quality. The same considerations apply to the distribution of music albums and, eventually, books. Traditional physical media carriers are substituted by mass-storage servers that take over the role of private CD collection and public library.

B-ISDNs allow information producers (such as film studios or record companies) to deliver their products directly to the client, eliminating the middleman. They also allow small information providers that are unable to compete with the distribution organization of today's large media companies to offer their products using the same infrastructure. Hence, the new technology creates both the danger of fostering information monopolies through centralization and the chance of democratizing the information market.

In today's home electronics and communications equipment we can identify a significant duplication of hardware due to disjoint ways of information handling and encoding. If all information has the same representation and is delivered in the same way, there is no longer a need for different terminal equipment in households and offices. In the future, a single computation, communication and entertainment (work-) station would suffice for a user who wants to access all services of the network. It would replace a personal computer, telephone, answering machine, FAX machine, TV receiver, VCR, and high-fidelity stereo set.

Here are the most immediate advantages for the user: avoiding redundant hardware for both the service provider and the consumer, in turn, making the provision of and participation in information services less costly than today. Furthermore, having a universal "media platform" provides more flexibility: new services do not require new equipment, but merely new software. Their establishment can proceed more rapidly – especially since the software can be obtained through the network as well.

2.2. Individual Information Selection

Once information receivers have computer functions available at their end of the information distribution channel, they have a high degree of control about the information they obtain [5]. Today newspaper agencies and TV stations choose which information they present to their customers and when they present it. With computers, users can select information themselves, at a time of their own choosing. Broadcast can be restricted to sending live information that is of interest to a large number of users at the same time, *e.g.*, a parliament session or a tennis match. All other information can be transmitted on demand; users stop a presentation, make a "detour" through background information, and return later to the interrupted program. Of course, not all users would like to change their current habits of information consumption and become involved in information services. The advantage of the computer lies in the adjustable degree of interaction; "couch potato mode" is always an option.

The powerful information retrieval abilities of the computer are essential for individual information access. Involvement of the user is not even needed: computers can be programmed to automatically filter out news in which the user has no interest. Knowing the user's preferences, the computer can also customize the information presentation, *e.g.*, when reporting the daily news to a sports fan, football results would go first, while for another political events will be the prime items. This will also make more "personalized" information services possible [6]: many people are more interested in news they receive through letters or electronic mail than in general news.

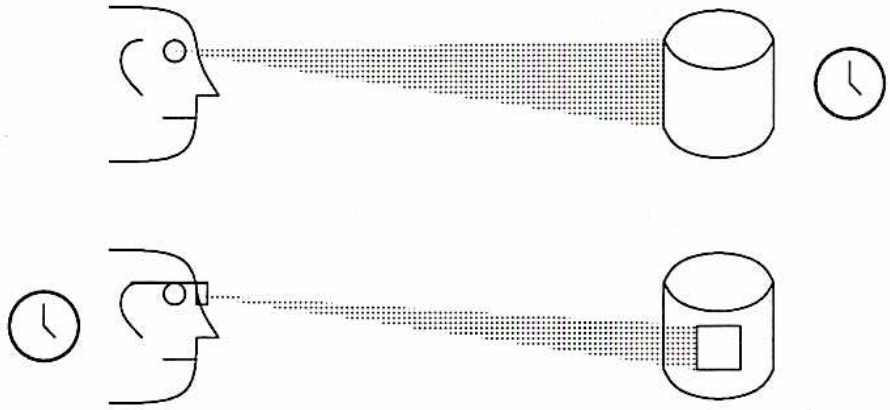

Figure 2: Information selection.
Above: Traditional broadcast at provider-determined times. Below: Individual choice.

A personalized information service would arrange all kinds of news items according to their importance to the user. Users can then access the information sequentially and will always receive the most important items first. Customized information services such as these are a major application field within artificial intelligence.

2.3. Flexible Information Presentation and Processing

Choosing the appropriate medium to present information is determined by whom and in what way the information is used. For a group of musicians, *e.g.*, the score (*i.e.*, a graphical representation) of the piece of music they want to play is most useful. Others would prefer to hear music than to read it. Another example: Instead of describing a car maintenance procedure in writing, the manufacturer could shoot a video showing the operation in practice, avoiding problems in correctly identifying car parts from the textual description. A third example: Today, airlines show videos demonstrating their plane evacuation procedures rather than relying on narration. It has been shown that people memorize the information much better this way, *i.e.*, the new type of media increases the usefulness of the presentation.

The more types of media a system is able to support, the better it can be adjusted to the needs of users and applications. An application area where the flexible combination of media is particularly promising is self-guided learning [7]. Students in a history class can, *e.g.*, browse through databases containing documents, maps, newsreels, TV documentaries, recordings of speeches, *etc.*, at their own convenience, guided by a *hypermedia system* [8, 9]. Another application field for multimedia systems is *computer-supported cooperative work* (CSCW) [10]. CSCW systems (colloquially termed "groupware") are tools for teams of users that work on a common project. They do not require the users to be physically present in the same place or at the same time, but allow them to use the same media they use for their cooperation now. Video conferencing [11] or voice mail [12] are today's primitive examples of such systems. Their expansion and integration to include different kinds of media will result in tools that support applications such as co-

Figure 3: Flexible media presentation.
Above: Graphical output. Below: Acoustic output.

authoring of multimedia documents or joint project management.

The ability to record and play back different kinds of media is the key element in supporting multimedia applications. The application range of a system, however, increases even further when different media can both be presented and also processed by the computer. A system's ability to "understand" audio, *e.g.*, makes it possible to issue spoken commands to it. This allows computers to be used by physically handicapped people or in situations where textual command input is impossible, *e.g.*, in a driving car.

One way in which a computer can process data is by transforming one information representation into another. This makes a flexible transition between different kinds of media possible: a movie can be generated from a textual description in the form of a program, a piece of music can be played from its score, *etc*. Today media transformation is mainly used in the production of video images for flight and car simulators, but the same computer animation techniques that imitate the real world as closely as possible can be used to create a "virtual reality" that follows different laws of nature. Perhaps there is no other area where the generality of the computer, its boundless abilities to handle and modify information, is as stimulating as in audio and video production [13]. While the computer cannot turn the average user into Paul McCartney or Walt Disney, it can at least remove technical and financial hurdles of creativity.

3. KEY ISSUES IN AUDIO AND VIDEO INTEGRATION

The integration of audio and video imposes new requirements on data processing systems. Audio and video data differs from text and graphics in its structure. Text and graphics are *discrete media* (DM): each data item constitutes a single (although arbitrarily complex) value. Audio and video, on the other hand, are *continuous media* (CM): their value changes continuously over time. (Note, that the terms "discrete" and "continuous" do not refer to the internal data representation, but to the users' view of the data. CM data can consist of a sequence of discrete values

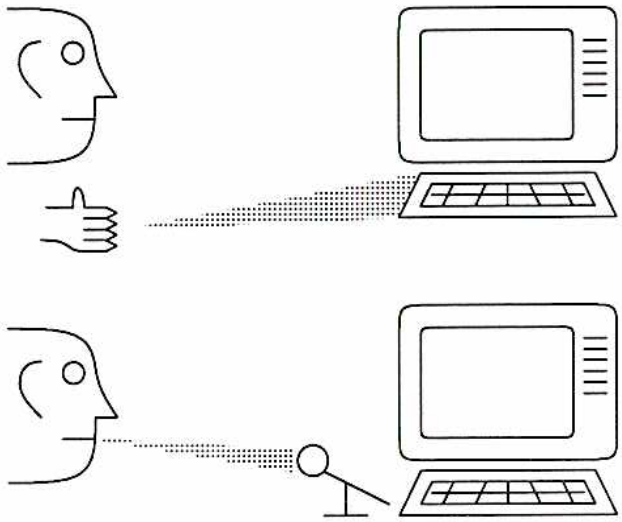

Figure 4: Flexible media processing.
Above: Textual input. Below: Voice input.

which replace each other as time goes on.) In the following subsections, we discuss the impact of these differences in data characteristics on the hardware, system software, and application interface of future multimedia systems.

3.1. Hardware Integration

Controlling CM with computers can be accomplished in a variety of ways. Using today's technology, the solution most readily available is not to abandon existing CM equipment such as CD players or VCRs, but to connect that equipment to the computer, executing control functions rather from some software module in the computer than from the operating panel of these devices. Initial systems featuring audio and/or video in the computing environment use this approach. An example is the Integrated Media Architecture Laboratory (IMAL) conceived at Bell Communications Research in Red Bank [14], an experiment in coordinating the provision of different media services offered through different communication utilities. Video services in the Pygmalion system of MIT's Project Athena [15] function in the same way: Each video workstation is connected both to an Ethernet for data communication and to a cable TV network to receive video signals. Through devices such as the Parallax video card [16], digital computer output and analog video information is combined so that it appears on the same screen.

In all these solutions, the problem is that the computer has a handle on the CM device rather than on the CM data. CM data does not enter the computer system; it passes through separate devices and its own communication lines. It cannot be manipulated by the computer; functions such as speech recognition are impossible in this approach. Furthermore, the granularity of control over CM data (*e.g.*, for synchronization purposes) is limited. The structure of such a system is similar to a process automation system in which the sensors and actors that connect the computer with

the technical process to be controlled impose uncertainties and delays [17].

Control of CM data can be more direct, and special hardware can be avoided, if the data passes through the computer system itself. The prerequisite for such a solution is that CM data has a digital representation. As we have argued in Section 2.1, future B-ISDN networks will deliver data in this way. In addition, user demands for quality improvements call for a digital representation because it allows data to be stored, copied, and exchanged without loss of signal quality. One of the first systems featuring digital audio in a computer environment was the Etherphone system developed at XEROX PARC [18] in which an Ethernet is used for data communication and telephony. Yet, with the exception of network communication, the Etherphone system keeps voice information and other data strictly separate by not allowing it to be loaded into the computer's main memory, by storing it on separate disks, *etc.*

Conceptually, digital CM I/O devices could be interfaced directly to the workstation bus. Today, however, the sheer volume of video data constitutes a major obstacle for this approach. Raw digital video information in traditional TV quality, *e.g.*, amounts to approx. 176 MBit of data every second. This data needs to be compressed before it can enter the computer system. Compression techniques, such as the one used with Intel's Digital Video Interactive (DVI) system [19], achieve a compression factor of approx. 150, albeit by tolerating some loss of video quality. In the DVI system, CM data can be stored in computer memory and can pass through the standard system bus. The same is possible with Digidesign's audio system for the Macintosh [20].

Systems such as the DVI system for the IBM PC [21] are intended for stand-alone local applications where contention for hardware resources is ruled out by design. In networked systems, where multiple concurrent applications involving the same workstation are supported by today's network-transparent window systems, contention may arise and will conflict with the performance requirements of CM data. To avoid such problems, Olivetti's Pandora's Box [22], developed together with the University of Cambridge, keeps compressed digital video data out of the workstation. This, of course, cures only the symptoms, not the cause. To allow compressed digital video to share standard system resources is not so much a question of capacity (which is likely to increase anyway), but a question of resource administration. It is a software rather than a hardware problem.

3.2. System Management

If concurrent processes handling CM and DM data share one machine, the operating system has to provide them with the system resources they need and to resolve resource conflicts. In traditional multitasking systems such as UNIX, "fairness" is the main criterion for resource administration. This criterion is inappropriate for handling CM. Apart from *throughput* requirements, CM applications impose *timing* demands on computer systems that result from the periodically changing value of CM data: each single value represents the CM data stream for some fraction of time. Changes in the times at which portions of CM data are played or recorded result in a modification of the original data semantics and must not happen unintentionally. To ensure correct timing, *delay* and *jitter* for the handling of CM data have to be bounded if some I/O equipment (and, obviously, some human user sitting in front of it) is involved in a CM application [23]. Without I/O (*e.g.*, when copying a video file), the handling of CM data is not time-critical.

To fulfill the timing requirements of CM data, the operating system should use *real-time scheduling* techniques. These techniques have to be applied to all system resources through which CM data passes, not just the CPU. Networks and disks can contribute more to the delay and jitter of CM data than processors; depending on the DMA capabilities of the devices CM data may not even pass through the CPU. To support the function of these schedulers, the deterministic behaviour of the operating system has to be ensured. Unpredictable effects of caching or page faults of a virtual memory system, *e.g.*, can ruin any carefully planned schedule.

Unfortunately, existing real-time systems are not well suited to support CM. Real-time scheduling is traditionally used only for *command and control systems* in application areas such as factory automation or aircraft piloting. For these applications, a large variety of real-time tasks, a plethora of I/O devices to interface with the technical process to be controlled, and high fault-tolerance requirements (that somewhat counteract to real-time scheduling efforts) are typical. CM applications have different (in fact, more favorable) real-time requirements:

- A sequence of digital CM data results from periodically sampling a sound or image signal. Hence, in processing the items of such a data sequence, all time-critical operations are periodic. Schedulability considerations for periodic tasks are much easier than for sporadic ones [24].
- Missing a deadline in a CM system is – although it should be avoided – not a severe failure. It may even be unnoticed: if a video frame (or parts of it) are not available on time it can simply be dropped. The human viewer will hardly notice it, provided this does not happen for a contiguous sequence of frames. For audio, requirements are higher because the human ear is more sensitive to audio gaps than the human eye is to video jitter.
- The fault-tolerance requirements of CM systems are usually less strict than for those real-time systems that have physical impact. The failure of a CM system will not directly lead to the destruction of technical equipment or constitute a threat to human life.

In a traditional real-time system, timing requirements result from the physical characteristics of the technical process to be controlled, *i.e.*, they are given externally. Some CM applications have to meet external requirements, too. Distributed music rehearsal is an example: music played by one musician on a instrument connected to his workstation has to be made available to all other members of the orchestra within a few milliseconds, otherwise they cannot keep a common time. If human users are involved in only the input or only the output of CM data, delay bounds are flexible. Consider the play-back of a video from a remote disk. How long it takes for a single video frame to be transferred from the disk to the monitor is unimportant to the user as long as frames arrive in a regular fashion. The user will notice any difference in delay only in the time it takes for the first video frame to be displayed. While the traditional real-time scheduling problem is to find a schedule for a set of processes with given delay bounds, often the problem in CM systems is to find reasonable delay bounds so that a set of processes is schedulable.

An appropriate way to reconcile an application's specific needs with a system's current possibility of accommodating real-time work items is to let both entities negotiate a "quality of service" immediately before this service is used [25]:

- Applications specify the workload they will impose on system resources and their performance requirements (if they have any) for the handling of this workload.
- In return, the operating system checks whether it can meet the requirements and, if so, provides performance guarantees and ensures meeting them as long as no hardware or software failure occurs and the application does not violate its workload specification.

Using this model, the operating system has control over the workload it accepts. It can refuse to service a new application if this application creates a workload that endangers the timing guarantees established for current users. Traditional real-time systems contain no mechanisms to turn down requests from the technical process to be controlled. In alarm situations, this can lead to an overload that causes the system to fail.

CM at the man-machine interface are an addition to – not a substitute for – the DM already available. In future multimedia systems, time-critical CM tasks and non-critical DM processes will run concurrently. Such a mixed operation is a new demand on scheduling; traditional systems usually have to support only one class of processes. The operating system must fulfill two conflicting goals:

- Time-critical processes must never be subjected to an unbounded *priority inversion* (*i.e.*, be kept from running by non-critical processes for an indefinite time).
- Uncritical processes should not suffer from *starvation* because time-critical processes are executed.

A solution to this conflict is possible because CM systems have control over the time-critical workload they accept. A fraction of the overall resource bandwidth can be set aside to serve non-critical processes. Under these new restrictions, the two-edged challenge to operating system designers remains to make the best use of system resources and to provide the best possible service.

3.3. Access to Continuous-Media Data

The benefits of letting CM data use standard system resources cannot be exploited by the user if CM data cannot be handled in the same software framework as other data types. Such a framework in today's distributed computer systems includes not only the services of the operating system, but also those of the communication network, the window system, and the programming toolkit. Well-established system paradigms such as I/O redirection and typical application tools such as mail should be applicable to CM data as they are to text. When graphics was introduced to computing, not much care was spent on this aspect of integration: it is still almost impossible today to send graphics mail and rely on its correct presentation to the receiver.

Object-oriented approaches seem to be best suited to model the integration of different media (see, *e.g.*, [26-30]). Their generic features and inheritance mechanisms make it feasible to apply standard system functions and application tools to a variety of data types. Yet, they take into account that different operations are applicable to each medium. In particular, they can handle the various presentation functions. For DM, presentation involves static data display. For CM, it requires a dynamic reproduction of the data sequence. In addition, the common graphical interface of object-oriented systems provides a uniform "look-and-feel" that makes these systems easy to use.

CM – by definition – elude the common *event feedback loop* of user input and system output. While a video is displayed the user needs to be able to issue commands to switch between channels or to change the volume. This is reflected best by a *multi-threaded* application structure where sporadic events and each periodically recurring operation are represented by their own threads. Hence, not only are CM applications used in a concurrent environment, they are inherently concurrent themselves.

The temporal aspects of CM data contribute to its semantics and are, therefore, not only important at the system management level, but also at the application interface. They result in a need for *synchronization* of threads that present CM data (to users or user processes). The following reasons for synchronization can be distinguished:

- Within a stream of CM data: The consecutive values of a CM stream must be displayed at regular intervals. For example, a new video frame in a TV service needs to be available every 33 ms.
- Between CM and DM data: If DM data is incorporated in CM data (as subtitles are in a movie) its processing or output has to occur when certain values of the CM data sequence are reached. If CM data is embedded in DM data (as in a voice-over text) one can either start its presentation automatically together with the DM data display or let the user initiate the presentation explicitly.
- Between different streams of CM data: If several streams of CM data are semantically connected, their values have to be presented together. A movie and its soundtrack, *e.g.*, have to be displayed in a way that synchronizes the spoken voice with the movement of the speaker's lips. Another example is the synchronization of two stereo audio channels.

Proposals to describe synchronization relations in multimedia systems can, among others, be found in [31, 32]. Synchronization can be expressed by arranging the presentation of DM and CM data on a common reference timeline (see, *e.g.*, [33]). In an actual implementation, all synchronization can then be achieved by explicit timing of operations. This makes the efficient management of fine-grain timers an important requirement for CM support, not only on the system, but also on the user level.

4. PROSPECTS

In the DASH Project at the International Computer Science Institute and the University of California at Berkeley we are currently in the process of designing and implementing a system to support CM. The project approach is called "*Integrated Digital Continuous Media*" (IDCM) [34] and stresses the issues outlined in the previous section. Instead of designing a new system from scratch, the DASH Project has decided to develop a set of abstractions needed to achieve IDCM and to incorporate these abstractions into existing widespread systems. This not only avoids having to deal with topics of no immediate relation to CM, but also provides the best chances of offering these solutions to large user communities.

On the system management level, our solutions focus on careful resource scheduling to meet the antagonistic demands of both DM and CM:

- We have developed the *DASH Resource Model* (DRM), a workload and scheduling model that provides a uniform abstraction for any system component from which CM services need to obtain real-time guarantees [35, 36]. The DRM can be realized for both preemptive resources such as the CPU [37] and for non-preemptive resources such as the network [38], taking into account that these resources not only have to serve regular CM workload, but also sporadic load from DM applications. We are currently integrating these schedulers into the Mach operating system.
- To apply the DRM to resources in distributed CM applications, we have defined the *Session Reservation Protocol* (SRP) that implements an end-to-end resource reservation algorithm in the context of TCP/IP networks [39]. It is used to provide guaranteed performance for traffic

on arbitrary IP-based connections. Again, SRP will be realized for the Internet protocol implementation of the Mach operating system [40].

On the application interface level, we have developed different solutions for presenting CM to the programmer and to the viewer or listener:

- As a programming interface for synchronized access to CM data, we have developed the abstraction of *Time Capsules* (TCs) [41]. TCs serve as container for CM data. Time parameters govern access operations to TCs and ensure regular reading and writing, complete with special read/write effects typical for CM data such as slow motion or time lapses.
- To record and reproduce CM data, we have designed *Abstractions for Continuous Media* (ACME) as a set of extensions to a network transparent window system for handling CM [42]. *Continuous Media Extensions to X* (CMEX) exemplify how these abstractions can be added as a backwards-compatible extension to the X window server [43].

Our current efforts concentrate on refining these solutions and getting them operational in a first IDCM prototype.

ACKNOWLEDGEMENTS

Michael Ahlsdorf inspired the writing of this paper by repeatedly asking for "the big picture". David Anderson, Ramesh Govindan, and George Homsy of the DASH Project have influenced the contents of this paper and provided valuable improvements. I also gratefully acknowledge the influence of discussions on my recent trip to Germany, in particular those at the University of Kaiserslautern, the German National Research Center for Computer Science in Karlsruhe, and the Technical University of Berlin. Special thanks go to Renee Reynolds for suggesting invaluable improvements to the style of this paper.

REFERENCES

1. "I-Series Recommendations", VIIIth Plenary Assembly, CCITT, 1985.
2. R. Haendel, "Evolution of ISDN Towards Broadband ISDN", *IEEE Network*, Jan. 1989, 7-13.
3. P. E. White, "The Broadband ISDN – The Next Generation Telecommunications Network", *Proceedings ICC*, 1986.
4. W. R. Byrne, T. A. Kilm, B. L. Nelson and M. D. Soneru, "Broadband ISDN Technology and Architecture", *IEEE Network*, Jan. 1989, 23-28.
5. K. A. Frenkel, "The Next Generation of Interactive Technologies", *Comm. of the ACM 32*, 7 (July 1989), 872-881.
6. S. Brand, *The Media Lab – Inventing the Future at MIT*, Viking Penguin Books, New York, 1987.
7. G. Blakowski, K. Coyle, J. Dirnberger, M. Duerr, M. Muehlhaeuser, B. Neidecker-Lutz, M. Richartz, T. Ruedebusch, J. Schaper, F. Spanachi, P. Tallett and I. Varsek, "NESTOR – Requirements and Architecture", Technical Report 13/89, University of Karlsruhe, Computer Science Department, Aug. 1989.

8. J. Conklin, "Hypertext: An Introduction and Survey", *IEEE Computer 20*, 9 (Sep. 1987), 17-41.

9. J. Smith and S. Weiss, "An Overview of Hypertext", *Comm. of the ACM 31*, 7 (July 1988), 816-819.

10. I. Greif, ed., *Computer-Supported Cooperative Work: A Book of Readings*, Morgan Kaufman Publishers, 1988.

11. S. Sarin and I. Greif, "Computer-Based Real-Time Conferencing Systems", *IEEE Computer 18*, 10 (Oct. 1985), 33-45.

12. R. H. Thomas, H. C. Forsdick, T. R. Crowley, R. W. Schaaf, R. S. Tomlinson, V. M. Travers and G. G. Robertson, "Diamond: A Multimedia Message System Built on a Distributed Architecture", *IEEE Computer*, Dec. 1985, 65-78.

13. W. E. Mackay and G. Davenport, "Virtual Video Editing in Interactive Multimedia Applications", *Comm. of the ACM 32*, 7 (July 1989), 802-810.

14. L. F. Ludwig and D. F. Dunn, "Laboratory for Emulation and Study of Integrated and Coordinated Media Communication", *Proc. of ACM SIGCOMM 87*, Stowe, Vermont, Aug. 1987, 283-291.

15. W. E. Mackay, W. Treese, D. Applebaum, B. Gardner, B. Michon, E. Schlusselberg, M. Ackermann and D. Davis, "Pygmalion: An Experiment in Multi-Media Communication", *Proceedings of SIGGRAPH '89*, Boston, MA, July 1989.

16. *The Parallax 1280 Series Videographic Processor*, Parallax Graphics, 1987.

17. R. G. Herrtwich and G. Hommel, *Kooperation und Konkurrenz – Nebenlaeufige, verteilte und echtzeitabhaengige Programmsysteme*, Springer-Verlag, Berlin, Heidelberg, New York, 1989.

18. D. B. Terry and D. C. Swinehart, "Managing Stored Voice in the Etherphone System", *Trans. Computer Systems 6*, 1 (Feb. 1988), 3-27.

19. G. D. Ripley, "DVI - A Digital Multimedia Technology", *Comm. of the ACM 32*, 7 (July 1989), 811-822.

20. W. Lowe and R. Currie, "Digidesign's Sound Accelerator: Lessons Lived and Learned", *Computer Music Journal 13*, 1 , 36-46.

21. A. C. Luther, *Digital Video in the PC Environment*, McGraw-Hill, 1989.

22. A. Hopper, "Pandora – An Experimental System for Multimedia Applications", *Operating System Review 24*, 2 (Apr. 1990), 19-34, ACM SIGOPS.

23. D. Ferrari, "Client Requirements for Real-Time Communication Services", Technical Report 90-007, International Computer Science Institute, Mar. 1990.

24. A. K. Mok, "The Design of Real-Time Programming Systems Based on Process Models", *IEEE Real-Time Systems Symposium*, Austin, Dec. 1984, 5-17.

25. R. G. Herrtwich and U. W. Brandenburg, "Accessing and Customizing Services in Distributed Systems", Technical Report 89-059, International Computer Science Institute, Oct. 1990.

26. A. J. Palay, "The Andrew Toolkit: An Overview", *Proceedings of the 1988 Winder USENIX Conference*, Dallas, Feb. 88, 9-21.

27. R. Steinmetz, "Synchronization Properties in Multimedia Systems", *IEEE Journal on Selected Areas in Communications 8*, 3 (Apr. 1990), 401-412.

28. T. D. C. Little and A. Ghafoor, "Synchronization and Storage Models for Multimedia Objects", *IEEE Journal on Selected Areas in Communications 8*, 3 (Apr. 1990), 413-427.

29. J. S. Sventek, *An Architecture Supporting Multimedia Integration*, Advanced Networked Systems Architecture Project, Technical Report AO.33.02, Jan. 1987.

30. E. Fiume and D. Tsichritzis, "Multimedia Objects", in *Active Object Environments*, D. Tsichritzis (editor), University of Geneva, June 1988, 121-128.

31. "Time Synchronization", IEC JTC1/SC18/WG3, N1443 (Working Paper), ISO, Oct. 1989.

32. "Representation and Protocols for the Exchange of Audiovisual Interactive Applications", IEC JTC1/SC18/WG3, N1428 (Working Paper), ISO, Sep. 1989.

33. J. S. Sventek and J. Wratten, *Temporal Aspects of Multimedia Data Structures*, Advanced Networked Systems Architecture Project, Technical Report TI.31.00, Feb. 1988.

34. D. P. Anderson, R. Govindan, G. Homsy and R. Wahbe, "Integrated Digital Continuous Media: a Framework Based on Mach, X11, and TCP/IP", Technical Report No. UCB/CSD 90/566, Mar. 1990.

35. D. P. Anderson, S. Tzou, R. Wahbe, R. Govindan and M. Andrews, "Support for Continuous Media in the DASH System", *Proc. of the 10th International Conference on Distributed Computing Systems*, Paris, May 1990.

36. D. P. Anderson and R. G. Herrtwich, "Resource Management for Digital Audio and Video", *IEEE Workshop on Real-Time Operating Systems and Software*, Charlottesville, May 1990, 99-103.

37. E. Barr, "CPU Scheduling in the DASH Resource Model", In preparation, June 1990.

38. M. Andrews, "Guaranteed Performance for Continuous Media in a General Purpose Distributed System", Masters Thesis, UC Berkeley, Oct. 1989.

39. D. P. Anderson, R. G. Herrtwich and C. Schaefer, "SRP: A Resource Reservation Protocol for Guaranteed-Performance Communication in the Internet", Technical Report 90-006, International Computer Science Institute, Feb. 1990.

40. D. P. Anderson, L. Delgrossi and R. G. Herrtwich, "Process Structure and Scheduling in Real-Time Protocol Implementations", Technical Report 90-021, International Computer Science Institute, June 1990.

41. R. G. Herrtwich, "Timed Data Streams in Continuous-Media Systems", Technical Report 90-017, International Computer Science Institute, May 1990.

42. D. P. Anderson, R. Govindan and G. Homsy, "Abstractions for Continuous Media in a Network Window System", In preparation, June 1990.

43. D. P. Anderson, R. Govindan and G. Homsy, "Continuous Media Extensions to X", In preparation, June 1990.