[HeSt91] *Ralf Guido Herrtwich, Ralf Steinmetz;* **Towards Integrated Multimedia Systems: Why and How**; Informatik-Fachberichte Nr.293, Springer-Verlag, 1991, S.327-342. Grundlage für den Zeitschriftenbeitrag [StHe93] im Informatik Spektrum, Springer-Verlag.

# Towards Integrated Multimedia Systems: Why and How

*Ralf Guido Herrtwich*
*Ralf Steinmetz*

IBM European Networking Center
Tiergartenstr. 8
D-6900 Heidelberg 1

**Abstract.** A multimedia system is characterized by the computer-controlled generation, manipulation, presentation, storage, and communication of independent discrete media such as text and graphics and continuous media such as audio and video. The innovation multimedia systems provide is flexibility through integration of different media into a single system. Such a system can unify the methods of information distribution, personalize information services through interactive access and individual information selection, enlarge the bandwidth of perception at the user interface, make information presentation more effective, and provide flexible media processing and transformation. The major obstacles to integrating continuous media with the discrete media in computer systems are limitations of today's digital solutions: In addition to high-capacity and high-speed hardware, system software is needed that meets the real-time demands of audio and video, and a multimedia application interface is compulsory. This paper elaborates on the key issues for providing multimedia support.

## 1. Introduction

The term "multimedia" has become one of the major buzzwords in computing and telecommunications for the 1990's [48; 50]. While everybody agrees that future systems should provide multimedia functionality, there is some uncertainty about what a multimedia system actually is and what its particular functions should be. Sometimes a product is called a "multimedia system" if it allows users to combine text and graphics. More ambitious people would not use this attribute before they can watch full-motion video in HDTV quality on their workstation screen while they are busy editing text files. The right answer is between these two extremes.

The purpose of this paper is to reasonably define the term and to examine the implications of this definition on future computer usage and computer system structures. The major challenges for building multimedia systems are identified. We investigate how issues that we consider crucial for multimedia systems are reflected in today's products or prototype systems and point out potential future developments. One assumption on which our discussion is based is the *distribution* of multimedia systems. Network attachment is a main feature of most computers today so that there is a strong requirement for multimedia functions to be applicable in a distributed environment as well.

Our discussion is based on the personal experiences we gained by working in two different multimedia research projects, DASH and DiME:

DASH is a project at the International Computer Science Institute and the University of California at Berkeley concerned with providing operating system support for digital audio and video [3]. The goal of the DASH project is to design and implement system software that allows audio and video to be handled just as any other kind of data in a typical workstation environment.

DiME is a project at the IBM European Networking Center Heidelberg dealing with distribution transparent access of multimedia resources like cameras and stored video sequences [52]. DiME aims to provide an "easy, but rich" communication service as part of an application programming interface to manipulate data streams by controlling their sources and sinks within a heterogeneous computing environment.

Apart from our own work, we take findings from recent multimedia conferences and workshops such as [5; 25; 40; 41; 49] into account.

The remainder of this text is organized in three main parts: Section 2 gives our definition of a multimedia system and provides the starting point of our discussion. Section 3 elaborates on the advantages of multimedia systems, identifying their potential for computing in general and for new application areas in particular. Based on this perspective, Section 4 as the core of the paper discusses the key issues faced in the design and realization of multimedia systems.

## 2. What is "Multimedia"?

Any information appealing to human senses is transported through some *medium*. From a computing perspective, media are means of communication between humans and computers - and between humans using computers as communication tools. Media represent information in particular ways: Whereas the medium "text" represents a formatted sentence visually as a sequence of characters, the medium "audio" represents it acoustically by means of pressure waves.

Each medium defines *presentation values* in a *presentation space* [19]: We are familiar with two-dimensional visual presentation space with color pixels on paper and on computer displays. But presentation spaces are not limited to the sense of vision: Stereophonic and quadrophonic sound define acoustic presentation spaces, and the sensors of our skin provide the presentation space for the sense of touch.

An additional presentation dimension for each medium is time. Some media, such as text and graphics, have time-independent values. These media are called *discrete*. Other media, such as audio and video, have values that change over time and these changes contribute to the media semantics. These media are called *continuous*. The terms "discrete" and "continuous" do not refer to the internal data representation, but to the users' view of the data. Continuous-media data often consist of a sequence of discrete values which replace each other as time progresses.

*Media flexibility* is the major requirement on a multimedia system.. To be flexible enough to handle all kinds of media, a multimedia system should *combine* both discrete and continuous media. With this requirement, neither a VCR nor a desktop publishing system handling text and graphics are multimedia systems, whereas an editor with voice annotation is. Yet, the mere incorporation of different media is not enough to achieve flexibility: A multimedia system should also be able to handle each type of media *independently*, providing the opportunity to combine them in arbitrary ways. A VCR, *e.g.*, stores audio and video together, prohibiting to access each information stream separately as it would be possible if both audio and video where contained in different files on a disk. To flexibly separate and combine different media, the computer is the ideal tool.

In summary, we arrive at the following definition [55]: *A multimedia system is characterized by the computer-controlled generation, manipulation, presentation, storage, and communication of independent discrete and continuous media.*

The individual components of multimedia systems do not necessarily have to be new and should not ignore existing techniques; we now have more than four decades of experience with discrete media in computing, and know how to handle voice and video communication through global networks. The innovation multimedia systems provide is in the ***integration*** of all kinds of media into a single system, obscuring the lines between computing, telecommunications, and even mass media. From a computing perspective, which is the natural starting point of our discussion, the major challenge of building a multimedia system is the introduction of continuous media into today's computer systems.

## 3. Potential of Integrated Multimedia Systems

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. With graphics, computers took over the function of traditional design tools, 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. Window system paradigms evolved together with the new media, and improved the way man and machine could communicate. The integration of audio and video will lead to corresponding paradigms. Therefore, the implications of multimedia systems are not just in opening new application domains, but in changing the view of computing in general.

Integrating continuous media in computer systems can offer a variety of advantages such as

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

With the availability of continuous media in distributed computer systems, potential advantages result from new possibilities for information distribution, selection, presentation, and processing. Since we should not design multimedia systems without these advantages in mind, we review the major potentials of integrated multimedia systems in the following subsections.

### 3.1 Unified Information Distribution

The telecommunications industry worldwide is moving rapidly towards the establishment of *integrated services digital networks* (ISDNs) [8], realizing that once all data has a common digital representation there is no need to maintain the existing multitude of channels for information distribution. The emerging optical technology together with advanced satellite communication leads to *integrated broadband communication networks* (IBCNs) [17; 6]: A single global multi-purpose network can carry not only text, data, and voice, but also video and high-fidelity audio, making it possible to offer high-quality video-conferencing and, on the long run, to provide television and radio services as well.

The high bandwidth and pervasiveness of future IBCNs make it possible not only to integrate existing communication networks, but also to replace and enhance traditional non-electronic channels of information distribution. The following example is typical for how the network can improve overall service quality: Instead of renting videos from a local video store, 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. The same considerations apply to the distribution of music albums and, eventually, books. Technically, traditional physical media carriers could be substituted by mass-storage servers that take over the role of private CD collection and public library. IBCNs will eventually allow information producers (such as film studios or record companies) to deliver their products directly to the client. This does not correspond to today's infrastructure for information distribution and implies severe economic changes.

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. Even in the future, there will not be only one universal audio or video format (HDTV, *e.g.*, seems to arise with three different standards). Format conversion services, however, will allow for the cooperation of various computer systems working with distinct data representations. Here, network integration pays off for the consumer: Avoiding redundant hardware for both the service provider and the consumer makes not only the provision of, but also the 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. With respect to the information distribution, users can electronically locate the best deal; there is no geographical limitation of a market, again implying economic change.

### 3.2 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 [15]. Today newspaper agencies and TV stations choose which information they present to their customers and when they broadcast it. With computers and telecommunication equipment, users can select information themselves, at a time of their own choosing: Videotex as a primitive example of such systems is also evolving towards the handling of multiple media [43]. 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 [7]: Many people are more interested in news they receive through letters or electronic mail than in general news. A personalized information service would arrange all kinds of news items according to their importance to the user. Customized information services such as these are a major application field within artificial intelligence [47].

Distributed multimedia systems will not replace books. Apart from constituting a cultural asset, printed material has the advantage of not requiring an electronic device for reading. Paper will always be attractive for entertaining reading. For news, its may loose some importance as electronic services can be more up-to-date. Scientific publishing is likely to be an area where distributed multimedia systems have the most severe impact. Both search and composition are facilitated by a computer system hooked to an information network.

### 3.3 Enlarged Bandwidth of Perception

Media determine not just *how*, but also *how good* man and machine can communicate. In the early days of computing, humans had to adapt to the computer for input and output of data. The introduction of graphical user interfaces and pointing devices has improved man-machine communication significantly. But still, today's forms of I/O are neither a very natural way for humans to communicate, nor are they very efficient: To speak is faster than to write, to listen is easier than to read, and to show is better than to describe. Audio and video increase the bandwidth of perception at the user interface of a computer system [33]: According to the idea transfer model [61] the *idea expression spectrum* (amount and range of information required to express a thought) is transformed with less loss of information from and to the *media spectrum* (amount of information expressed using different media).

Choosing the appropriate medium to present information is determined by whom and in what way the information is used. With a variety of media available, it becomes possible to adjust the "look and feel" of computer-based work to familiar human operations. Proof for the importance of such adjustment is the success of the desktop paradigm used in today's window systems. To model sheets of paper on a computer screen is tightly coupled with the availability of appropriate media (graphics, in this case). Corresponding paradigms will evolve for systems incorporating audio and video.

The more 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 [39]. 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* [11; 42]. Another application field for multimedia systems is *computer-supported cooperative work* (CSCW) [16]. 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 [53] or voice mail [60] 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-authoring of multimedia documents or joint project management.

### 3.4 Flexible Media Processing and Transformation

Not only information access, but also information production is easier with a distributed multimedia system. While this is beneficial at an individual level, e.g., for the author of a paper, it can be questioned whether it is good for the society as a whole. The easier it is to put together a piece, the less important it is whether the content is worth the effort. Everything can be put together so fast that the amount of worthless information increases. The ratio of good versus bad pieces can become worse than today.

The ability to record and play back information coded in different kinds of media is the key element in supporting multimedia applications. The application range of a system, however, increases even further when these media can also be 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, 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 [35]. While the computer cannot turn the average user into George Gershwin or Walt Disney, it can at least remove technical and financial hurdles of creativity.

## 4. Key Issues in Integrated Multimedia System Design

We have identified *flexibility through integration* as the major feature of a multimedia system. This integration needs to take place throughout the entire system, *i.e.*, at the *hardware adaptation*, the *system management* and the *application interface level.* The integration issues, however, are different on each level as discussed in the following subsections.

### 4.1 Hardware Adaptation Level

Continuous and discrete media are traditionally handled in completely separate environments. As shown in Figure 1, exchange of continuous media takes place in analog form, involving very little processing. Even if in today's TV and radio studios many components handling continuous media become digital, analog signals are still switched between the various sources and sinks. Discrete media, on the contrary, are handled in a completely digital environment, both for processing and exchange.

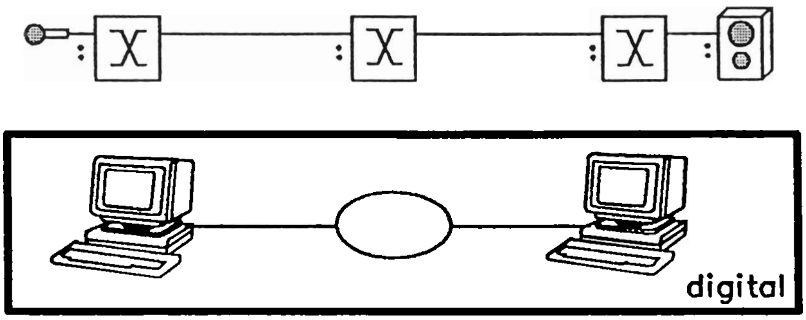


**Figure 1. Disjoint analog and digital data paths**

#### 4.1.1 Analog Technology as a Starting Point

Using today's technology, the solution most readily available to integrate continuous and discrete media is not to abandon existing continuous-media equipment such as CD players or VCRs, but to connect this equipment to the computer via some interface (e.g., RS-232, RS-424, SCSI). Control functions are then executed rather from some software module in the computer than from the operating panel of these devices. This is shown in Figure 2. As a consequence, the representation of continuous media in such a system is determined by the existing devices. While for audio devices digital encoding can be used, video in such system is usually available in analog form only.

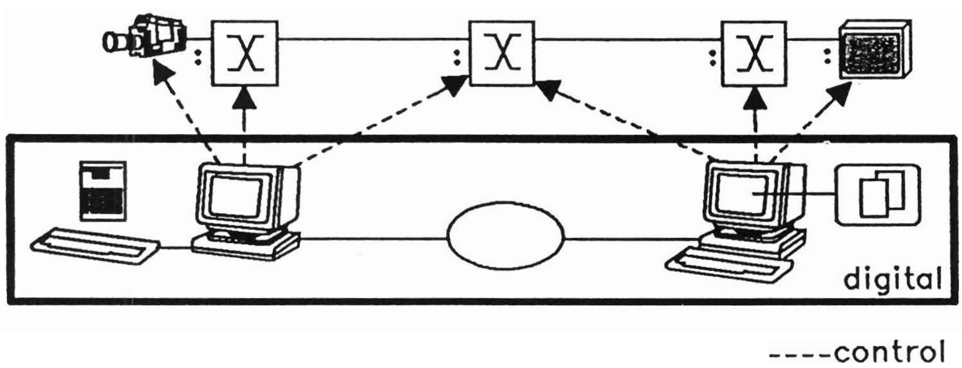


**Figure 2. Computer controls all continuous-media devices**

An example of a system using this approach is the Integrated Media Architecture Laboratory (IMAL) conceived at Bell Communications Research in Red Bank [32]. IMAL is an experiment in coordinating the provision of a variety media services offered through different communication utilities. Video services in the Muse and Pygmalion system of MIT's Project

Athena [22; 10; 38] function in the same way: Each video workstation is connected both to an Ethernet for traditional data communication and to a cable TV network to receive video signals. Through devices such as the Parallax video card [45], digital computer output and analog video information is combined so that it appears on the same screen.

The first experiments within the DiME project [52] at the IBM European Networking Center were performed on a similar basis using PS/2 workstations with AVC and M-Motion adapters [4; 36]. Devices generating or processing continuous media can be controlled from remote workstations. DiME, however, aims to use fully digital communication between various workstations as shown in Figure 3. Since devices receiving or generating analog signals may be used as local attachments, some digitization needs to take place before data can enter the digital network. A similar experiment took place at US West in Boulder [12] where a DS3 optical fiber interconnects two sites with their multimedia labs. Within the sites all continuous-media signals are interconnected using analog switches.

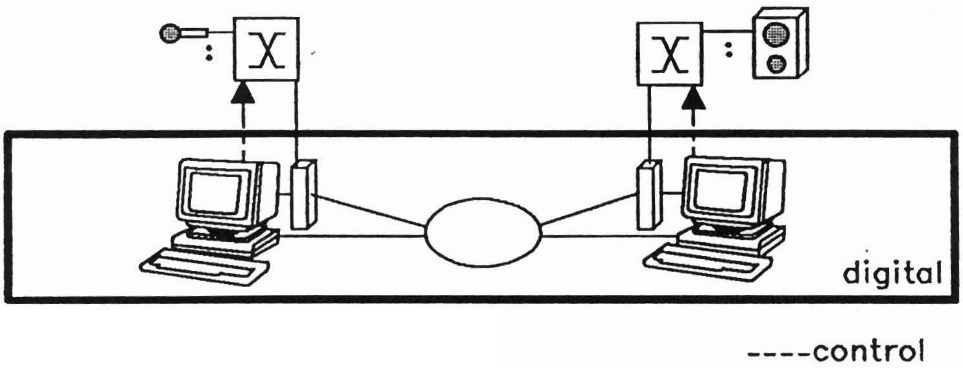


**Figure 3. Local continuous-media devices attachments, nonintegrated digital continuous-media communication**

### 4.1.2 Fully Digital Systems as the Final Goal

A main advantage of approaches utilizing analog technology is their feasibility today: Devices are available to handle audio and video in real time. They provide a testbed for experiments such as user interface and application studies for which the underlying technology is of minor importance. Demands for integrated solutions as described in Section 3 call for digital data representations. Digital encoding also allows for quality improvements since data can be stored, copied, and exchanged without loss of signal quality.

The above-mentioned approaches have the problem that the computer handles continuous-media device rather than the continuous-media data. This data does not enter the computer system after being generated; it passes through separate devices and its own communication lines. It cannot be manipulated directly by the computer; functions such as speech recognition are only possible in this approach if the signals are fed somewhere into the computer. Furthermore, the granularity of control over continuous media (*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 impose uncertainties and delays.

Intermediate solutions operate with analog devices and some analog data streams for continuous media within the local environment (see Figure 4). The hybrid data handling remains; it imposes functional restrictions and adds complexity. Data from local analog devices is routed through a special local switch. It is treated differently from a digital media stream derived, *e.g.*, from the fixed disk and delivered to a video window.

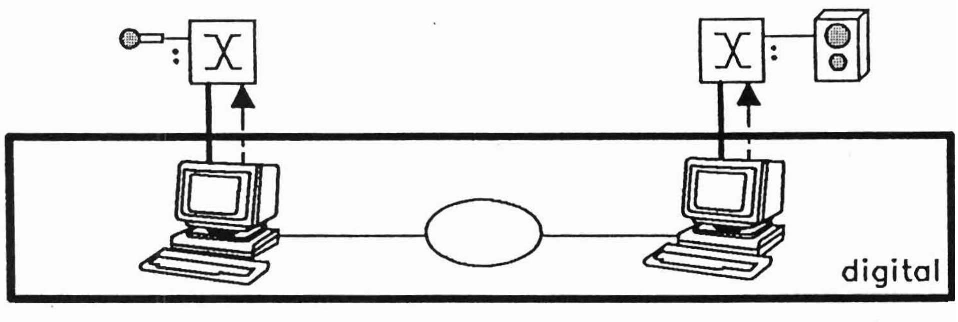


**Figure 4. Local continuous-media devices attachments, digital continuous-media communication**

The control of continuous media can be more direct if all data passes through the computer system itself. This is only possible with digital data encoding (where it is of secondary importance whether data is digitized in the workstation as shown in Figure 5 or in the I/O device. One of the first systems featuring digital audio in a computer environment was the Etherphone system developed at XEROX PARC [59] 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. A similar approach was used in a project by AT&T in Naperville [27; 28]: 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. This was demonstrated by building a conferencing application.

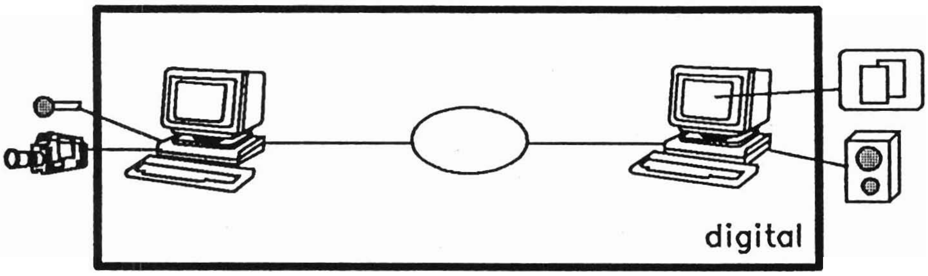


**Figure 5. Unified approach**

Conceptually, digital continuous-media I/O devices could be interfaced directly to the workstation bus once they have digital interfaces. At least for video, however, the sheer amount of data constitutes the major obstacle to transporting and storing it in raw digital form: A digital TV signal in traditional studio quality encoded according to CCIR 601, *e.g.*, requires 216 Mbit/s, an HDTV signal (e.g., HD-MAC) requires 216 Mbit/s * 5.33 $\simeq$ 1.152 Gbit/s [57]. Even higher data rates occur at the production of these HDTV signals.

Video 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 [34], achieve a compression factor of appr. 150, albeit by tolerating some loss of video quality. In the DVI system, continuous media 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 [31]. With algorithms such as MPEG developed at ISO/IEC JTC1/SC2 [63] or DVI low TV quality is possible at data rates around 1.4 Mbit/s.

Digital systems provide the additional advantage of increased flexibility. Today an audio mixer is typically a piece of hardware, but its functions could also be implemented in software running on a digital signal processor. The advantage of the software solution is that it can easily be adapted to changing requirements. The DVI system provides a good example: Its video display processor VDP1 (the pixel processor) is microcoded. Its code, which is part of the video compression/decompression chain, is loaded during the initialization phase, leaving the door open for future improvements of the compression algorithm. While performance requirements may not always allow software solutions, digital systems offer the possibility of continuously adjusting the border between hardware and software implementations based on the ratio of performance over processing power, cost and flexibility.

### 4.2 System Management Level

Available systems using DVI or the Compact Disc Interactive system as a whole (CD-I was developed mainly by Philips and published as the "Green Book" standard in a joint effort of Philips and Sony) [9] are intended for stand-alone local applications. In these systems, contention for hardware resources is ruled out by design. In multi-process/multi-user or even networked systems, where multiple concurrent applications on the same workstation are supported, contention may arise and will conflict with the performance requirements of continuous media. To avoid such problems, Olivetti's Pandora's Box [21], 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, but a question of resource administration. It is a system management rather than a hardware problem.

#### 4.2.1 Real-Time Scheduling

If concurrent processes handling continuous and discrete media 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 insufficient for handling continuous media. Apart from high *throughput* requirements, continuous media impose *timing* demands on computer systems that result from the periodically changing value of continuous-media data: Each single value in an audio or video stream represents stream information for some fraction of time. Changes in the times at which values 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 continuous media have to be bounded if some I/O equipment (and, obviously, some human user sitting in front of it) is involved in the application [13]. Without I/O (*e.g.*, when copying a video file), the handling of continuous media is not time-critical.

To fulfill the timing requirements of continuous media, the operating system must use *real-time scheduling* techniques. These techniques have to be applied to all system resources through which continuous-media data passes, *i.e.*, to the entire *end-to-end* data path, not just the CPU. Networks and disks can contribute more to delay and jitter than processors. With DMA capabilities of controllers, continuous-media data may not even pass through the CPU. To support the function of these schedulers, the deterministic behavior of the operating system has to be ensured. Unpredictable effects of caching, process switches 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 continuous media. Real-time scheduling is traditionally used 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. Continuous media have different (in fact, more favorable) real-time requirements:

- A sequence of digital continuous-media 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 [37].

- For many applications missing a deadline in a multimedia system is - although it should be avoided - not a severe failure. It may even be unnoticed: If an uncompressed 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 continuous-media systems are usually less strict than for those real-time systems that have physical impact. The failure of a continuous-media system will not directly lead to the destruction of technical equipment or constitute a threat to human life. Of course, multimedia systems should be reliable, but not more or less than traditional data processing systems.

- The bandwidth demand of continuous media is not always *that* stringent. As some compression algorithms are capable of using different compression ratios - leading to different qualities - the required bandwidth can be negotiated. If not enough capacity for full quality is available the application may also accept a reduced quality (instead of no service at all). The quality may also be adjusted dynamically to the available bandwidth, *e.g.*, by changing encoding parameters.

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 continuous-media applications have to meet external requirements, too. Distributed music rehearsal is an example: Music played by one musician on an 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 continuous media, 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 multimedia systems is to find reasonable delay bounds so that a set of processes is schedulable.

Continuous media are an addition to - not a substitute for - the discrete media already available. In future multimedia systems, time-critical continuous-media tasks and non-critical discrete-media 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 *priority inversion* (*i.e.*, be kept from running by non-critical processes for an indefinite time) [51].

  Uncritical processes should not suffer from *starvation* because time-critical processes are executed.

A solution to this conflict is possible if multimedia 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. A scheduling system based on these considerations is used in the DASH system [2] developed at the International Computer Science Institute and the University of California at Berkeley.

#### 4.2.2 Quality of Service Management

An appropriate way to reconcile an application's specific needs with a system's current possibility of accommodating work items is to let both entities negotiate a *"quality of service"* immediately before this service is used [18]:

- 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 (for throughput, loss and delay) 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. Instead, they have to take into account exception handling procedures for alarm situations.

To guarantee a certain quality of service to an application, it is common that the system reserves a fraction of the overall system capacity exclusively for this application. Usually such a reservation is *pessimistic* [2]: It is made for the worst case, *i.e.*, for the largest workload. If the actual workload can differ significantly from the worst case (as, *e.g.*, in variable bit-rate encoding schemes) this may lead to an underutilization of resources. Service requests may also be turned down needlessly. An alternative is *optimistic* reservation [20] where requests are only turned down if they cannot even be tolerated in the best-possible case. Unlike the pessimistic scheme, optimistic reservation cannot avoid that violations of service quality occur. It, therefore, has to provide methods for *quality-of-service monitoring* (*e.g.*, detecting the violation of deadlines for the delivery of continuous-media messages) and *conflict resolution* (*e.g.*, by aborting the least important application). Both schemes have their applications: Pessimistic reservation is needed for video production and high-fidelity audio playback; optimistic methods are appropriate for voice communication, video conferencing and television services.

Quality-of-service negotiation is usually thought of as transferring a request with a fixed set of parameters from the application to the system. These parameters will then either be accepted or refused. This *imperative* approach can be extended to a more *cooperative* negotiation scheme: The application can define its preferred qualities and a range of other qualities that are also acceptable; the system can then flexibly reconcile the application's needs with its own possibilities. Such a quality-of-service negotiation can become arbitrarily complex, in particular if different parameters depend on each other. In deciding on a quality-of-service, not only the usefulness of a service quality (indicated, *e.g.* by a *service metric* as described in [18]), but also the *service cost* could be used. Again, the DASH project has addressed these issues [1].

### 4.3 Application Interface Level

The benefits of letting continuous media media pass through standard system resources cannot be exploited by the user if continuous media 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 and communication system, but also the window system, the programming toolkit, and the data model. Well-established user interface paradigms such as I/O redirection and typical application tools such as mail should be applicable to continuous media 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.

In addition to new mechanisms needed at the application interface level (discussed below), existing problems in any distributed programming environment become more severe in an integrated multimedia system. These problems include heterogeneity (caused by connecting a large number of different I/O devices to the system), transparency (resulting from the need to handle multimedia objects at different locations) and protection (required to prevent unauthorized media usage).

### 4.3.1 Program and Data Model

*Object-oriented* approaches seem to be suited best to model the integration of different media. Almost all program models for multimedia systems use them (*e.g.*, [14; 29; 44; 54; 56; 58]). Often multimedia devices are represented through a class hierarchy with common operations on a generic device as the basic class. Subclasses may be input, output, in/out or storage devices. A camera and a microphone are, *e.g.*, subclasses of input devices. Other approaches try to establish a media hierarchy with operations common to the specific medium. A more application-oriented solution can focus on a representation of, *e.g.*, a book as the paradigm for documents and build subclasses for types or components of books such as video clips.

Generic features and inheritance mechanisms in object-oriented models 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 by a unified form. For discrete media, presentation involves static data display. For continuous media, 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.

Continuous media — 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 multimedia applications used in a concurrent environment, they are inherently concurrent themselves.

Inter-thread communication will look different for discrete and continuous media. Whereas for discrete media single data values are transferred, for continuous media the application should just need to define sources and sinks once and then continuously transfer data. In addition to "read" and "write", "connect/start" and "disconnect/stop" operations are required [27].

In regard to the data model, enhancements and/or adaptions of existing models such as SGML [24] or ODA [23] are desirable. SGML provides the basis for multimedia specific enhancements like SDML and HyTime, or may even be used as a framework for other information architectures [26]. In contrast to SGML, ODA has predefined semantics to specify the structure and the layout of documents which must be enhanced towards multimedia capabilities [19].

### 4.3.2 Synchronization

The temporal aspects of continuous-media 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 continuous-media and discrete-media data (to users or user processes). The following reasons for synchronization can be distinguished:

- *Between different continuous-media streams:* If several continuous-media streams 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.

- *Between continuous-media streams and discrete-media data items:* If discrete-media data items are incorporated in continuous-media streams (as subtitles are in a movie) their processing or output has to occur when predefined events or time stamps of the continuous-media stream are reached. If embedding works the other way round (as in voice-over text) one can either start its presentation automatically together with the discrete-media data display or let the user initiate the presentation explicitly.

Whether the user can influence the synchronization that shall take place depends on the application. Consider the following two examples:

1. A camera and a microphone are attached to a workstation on which a joint editing application runs as part of a video conference. While speaking, the owner of a shared window points to various graphical objects. At the remote workstation the audio and video data must be presented simultaneously and synchronized with the pointer.

2. Surrogate travel applications such as the Aspen Movie Map of the MIT or a tour through the Palenque ruins [62] allow users to move electronically through a new building: The user defines direction and actions to be performed like "turn right and open door" by pressing appropriate buttons on the graphical user interface. Then the sound of the opening door and a video showing the walk through the room are presented. Additional information may appear as overlay text from time to time.

The first example involves *live synchronization*: Synchronous input of data shall result in synchronous output. Live synchronization should happen automatically; the user is not involved. In the second example, data does not evolve on the fly, but is available beforehand. In this case, which is called *synthetic synchronization* [30], users can freely order the various data entities in the time domain. For this purpose, they need to be able to express their synchronization requirements through corresponding language constructs. Notions like "present data entity A simultaneously/after/independently from data entity B" are needed [46]. These constructs apply either to the whole information entity or may refer to time or event stamps within the entity [56]. Within an integrated multimedia system both types of synchronization are required.

## 5. Final Remark

In the previous sections, we have identified the main criteria for the development of integrated multimedia systems. Looking at today's products, we find that they are still far away from having accomplished the goal of integration. If we, *e.g.*, look at CD-I, one of the most advanced technologies for multimedia retrieval, we notice that it is only provided within closed systems intended for this single application. If we look at the NeXT station which has pioneered audio in the workstation environment, we find that parallel audio output and mouse movements interfere with each other, causing annoying sound glitches. Yet, such systems are essential to understand the requirements of multimedia applications and to experiment with different solutions. Today's systems are beneficial as long as we keep in mind that they do not yet represent the future of information processing and exchange — they are, however, leading the way.

**Acknowledgement**

We would like to thank our colleagues within the DASH and DiME projects, in particular David Anderson of the University of California at Berkeley and Johannes Rückert, Bernd Schöner and Hermann Schmutz of the IBM European Networking Center Heidelberg, who have contributed indirectly to this paper by many - often very controversial – discussions.

**References**

[1] *David P. Anderson, Ralf Guido Herrtwich, Carl Schaefer;* **SRP: A Resource Reservation Protocol for Guaranteed-Performance Communication in the Internet;** International Computer Science Institute, Berkeley, Technical Report 90-006, Feb. 1990.

[2] *David P. Anderson, Ralf G. Herrtwich;* **Resource Management for Digital Audio and Video;** IEEE Workshop on Real-Time Operating Systems and Software, Charlottesville, May 1990, pp. 99-103.

[3] *D.P. Anderson, S. Tzou, R. Wahbe, R. Govindan, M. Andrews;* **Support for Continuous Media in the DASH System;** Proc. of the 10th International Conference on Distributed Computer Systems, Paris, May 1990.

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

[5] **International Workshop on Network and Operating System Support for Digital Audio and Video;** International Computer Science Institute (ICSI), Berkeley, CA, Nov. 8-9, 1990.

[6] *William R. Byrne, Toomas A. Kilm, Bruce L. Nelson, Marius D. Soneru;* **Broadband ISDN Technology and Architecture;** IEEE Network Magazine, vol.3, no.1, Jan. 1989, pp. 23-28.

[7] *Steward Brand;* **The Media Lab, Inventing the Future at MIT;** Viking Penguin, 1987.

[8] *The International Telegraph and Telephone Consultative Committee.* **I-Series Recommendations;** VIIIth Plenary Assembly, Melbourne 1988.

[9] *Philips International B.V.;* **Compact Disc-Interactive - A Designers Overview;** Kluwen, Technische Boeken B.V., Deventer, Netherlands, 1989.

[10] *Georg Champine, Daniel Geer, William Ruh;* **Project Athena as a Distributed Computer System;** IEEE Computer, vol.23 no.9, September 1990, pp.40-51.

[11] *J. Conklin.* **Hypertext: A Survey and Introduction.** IEEE Computer, Sep. 1987, pp. 17-41.

[12] *Doug Corey, Jill Schmidt, Mark Abel, Stephen Bulick, Steve Coffin;* **Multimedia Communications: The US West Advanced Technologies Prototype Telecollaboration System;** 5th IEEE International Workshop on Telematics, Denver, Colorado, USA, Sept. 17-21, 1989.

[13] *Domenico Ferrari;* **Client Requirements for Real-Time Communication Services;** International Computer Science Institute, Technical Report 90-007, Berkeley, March 1990.

[14] *E. Fiume, D. Tsichritzis;* **Multimedia Objects;** in: Active Object Environments, D. Tsichritzis (Ed.), University of Geneva, June 1988, 121-128.

[15] *Karen A. Frenkel;* **The Next Generation of Interactive Technologies;** Communications of the ACM, vol.32, no.7, Jul. 1989, pp. 872-881.

[16] *Irene Greif;* **Computer-Supported Cooperative Work: A Book Of Readings;** Morgan Kaufmann Publishers, 1988.

[17] *Rainer Händel;* **Evolution of ISDN Toward Broadband ISDN;** IEEE Network Magazine, vol. 3, no.1, Jan. 1989, pp. 7-13.

[18] *Ralf Guido Herrtwich, Uwe Wolfgang Brandenburg;* **Accessing and Customizing Services in Distributed Systems;** International Computer Science Institute, Technical Report 89-059, Berkeley, Oct. 1989.

[19] *Ralf Guido Herrtwich, Luca Delgrossi* **ODA-Based Data Modeling in Multimedia Systems** International Computer Science Institute, Technical Report 90-043, Berkeley, 1990.

[20] *Ralf Guido Herrtwich;* **The DASH Resource Model Revisited or Pessimism Considered Harmful**; International Workshop on Network and Operating System Support for Digital Audio and Video; International Computer Science Institute, Berkeley, Nov. 8-9, 1990.

[21] *Andy Hopper;* **Pandora - An Experimental System for Multimedia Applications**; ACM, Operating Systems Review, Apr. 1990, pp. 19-34.

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

[23] *ISO;* **Information Processing - Office Document Architecture and Interchange Format**; ISO, Genf 1989.

[24] *ISO;* **Information Processing - Standard Generalized Markup Language**; ISO, Genf 1986.

[25] **5th IEEE International Workshop on Telematics**; Denver, Colorado, USA, September 17-21, 1989.

[26] *Jürgen Kämper;* **Towards Extensible Multimedia Document Architectures**; IEEE Multimedia '90, Bordeaux, Nov. 15-17., 1990.

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

[28] *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, Apr. 1990, pp. 380-390.

[29] *T.D.C. Little, A. Ghafoor;* **Synchronization and Storage Models for Multimedia Objects**; IEEE Journal on Selected Areas in Communication, vol.8, no.3, Apr. 1990, pp. 413-427.

[30] *Thomas D.C. Little, Arif Ghafoor;* **Network Considerations for Distributed Multimedia Objects Composition and Communication**; IEEE Network Magazine, vol.4 no.6, Nov. 1990, pp. 32-49.

[31] *W. Lowe, R. Currie* **Digidesign's Sound Accelarator: Leasons Lived and Learned** Computer Music Journal 13, 1, 36-46.

[32] *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, Aug. 11-13,1987.

[33] *V.Y. Lum, K. Meyer-Wegener;* **An Architecture for a Multimedia Database Management System Supporting Content Search**; in: Proc. Int. Conf. on Computing and Information, Niagara Falls, Ontario, Canada, May 23-26, 1990.

[34] *Arch C. Luther.* **Digital Video in the PC Environment.** Intertext Publications McGraw-Hill Book Company New York.

[35] *Wendy E. Mackay, Glorianna Davenport;* **Virtual Video Editing in Interactive Multimedia Applications**; Communications of the ACM, vol.32, no.7, Jul. 1989, pp. 802-810.

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

[37] *Al K. Mok;* **The Design of Real-Time Programming Systems Based on Process Models**; IEEE Real-Time Systems Symposium, Austin, Dec. 1984, pp. 5-17.

[38] *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.

[39] *M. Mühlhäuser;* **Requirements and Concepts for Networked Multimedia Courseware Engineering**; International Conference on Computer Aided Learning 89, 1989, Dallas, USA.

[40] **2nd IEEE COMSOC International Multimedia Communications Workshop**; Montebello, Quebec, Canada, Apr. 1989.

[41] **3rd IEEE COMSOC International Multimedia Communications Workshop**; Bordeaux, France, Nov. 15-17, 1989.

[42] *Jakob Nielsen;* **Hypertext and Hypermedia**; Academic Press, 1990.

[43] *F.Oguet, C.Schwarts, F.Kretz, M.Quere;* **RAVI, A Proposed Standard for the Exchange of Audio/Visual Interactive Applications**; IEEE Journal on Selected Areas in Communication, vol.8, no.3, April 1990, pp.428-436.

[44] *A.J. Palay* **The Andrew Toolkit: An Overview** Proceedings of the 1988 Winter USENIX Conference, Dallas, Feb. 1988, 9-21.

[45] *Parallax Graphics;* **The Parallax 1280 Series Videographic Processor**; 1987.
[46] *Jonathan B. Postel, Gregory G.Finn, Alan R. Katz, Joyce K.Reynolds.* **An Experimental Multimedia Mail System.** ACM Transactions on Ofiice Information Systems, vol.6, no.1, Jan. 1988, pp. 63-81.
[47] *Stephen Pollock;* **A Rule-Based Message Filtering Sysytem**; ACM Transactions on Office Information Systems, vol. 6, no. 3, Jul. 1988, pp. 232-254.
[48] *Larry Press;* **Computer or Teleputer?**; Communications of the ACM, vol. 33, no. 9, September 1990, pp. 29-36.
[49] **Real-Time Systems Symposium**; Orlando, December 5-7, 1990.
[50] *Phillip Robinson;* **The Four Multimedia Gospels**; Byte, vol.15, no. 2, Feb. 1990, pp. 203-212.
[51] *R. Rajkumar, L. Sha, J. P. Lehoczky;* **Real-Time Synchronization Protocols for Multiprocessors**; IEEE Real-Time Systems Symposium, Huntsville, December 6-8, 1988, pp. 259-269.
[52] *Johannes Rückert, Hermann Schmutz, Bernd Schöner, Ralf Steinmetz;* **A Distributed Multimedia Environment for Advanced CSCW Applications**; IEEE Multimedia '90, Bordeaux, Nov. 15-17, 1990.
[53] *Sunil Sarin, Irene Greif;* **Computer-Based Real-Time Conferencing Systems**; IEEE Computer, vol. 18, no. 10, Oct. 1985, pp. 33-45.
[54] *Ralf Steinmetz, Reinhard Heite, Johannes Rückert, Bernd Schöner;* **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, Nov. 8-9, 1990
[55] *Ralf Steinmetz, Johannes Rückert, Wilfried Racke;* **Multimedia-Systeme**; Informatik Spektrum, Springer Verlag, vol.13, no.5, 1990, pp.280-282.
[56] *Ralf Steinmetz;* **Synchronization Properties in Multimedia Systems**; IEEE Journal on Selected Areas in Communication, vol. 8, no. 3, April 1990, pp. 401-412.
[57] *L. Stenger;* **Digital Coding of Television Signals-CCIR Activities for Standardization**; Signal Processing Image Comm. vol. 1 nr. 1, Jun. 1989, pp. 29-43.
[58] *J.S.Sventek;* **An Architecture for Supporting Multimedia Integration**; IEEE Computer Society Office Automatition Symposium, Apr. 1987, pp.46-56.
[59] *Daniel C. Swinehart;* **Telephone Management in the Etherphone System**; IEEE Globecom'87, 1987, pp. 30.3.1-30.3.5.
[60] *Robert H. Thomas, Harry C. Forsdick, Terrence R. Crowley, Richard W. Schaaf, Raymond S. Tomlinson, Virginia M. Travers;* **Diamond: A Multimedia Message System Built on a Distributed Architecture**; IEEE Computer, Dec. 1985, pp. 65-78.
[61] *Hitoshi Watanabe;* **Integrated Office Systems: 1995 and Beyond**; IEEE Communication Magazine, vol. 25, no. 12, Dec. 1987, pp. 74-80.
[62] *K. S. Wilson;* **Palenque: An Interactive Multimedia Optical Disk Prototype for Children**; Bank Street College of Education, NY, Technical Report 2, 1987.
[63] *Hiroshi Yasuda;* **Standardization Activities on Multimedia Coding in ISO**; Signal Processing: Image Communication, vol. 1, no. 1, 1989, pp.3-16.

J. Encarnação (Hrsg.)

# Telekommunikation und multimediale Anwendungen der Informatik

GI-21. Jahrestagung
Darmstadt, 14.-18. Oktober 1991

Proceedings

Springer-Verlag
Berlin Heidelberg New York London Paris
Tokyo Hong Kong Barcelona Budapest