# Tools for Digital Lecturing
## - What We Have and What We Need -

Wolfgang Effelsberg, Werner Geyer
University of Mannheim
Germany
http://www.informatik.uni-mannheim.de/informatik/pi4

**Abstract**

In the past many distance learning projects have concentrated on the development of educational multimedia material for offline use. Effective learning requires online contact between teachers and students, and also among students. In this paper we claim that the multimedia application software we have today is insufficient for online (synchronous) teleteaching. We list the requirements for good teleteaching tools, analyze the deficiencies of existing software for the purpose of teleteaching, and present the teleteaching tools currently under development at the University of Mannheim.

## 1. Introduction

In recent years many universities worldwide have begun to offer multimedia courseware over their campus networks; some are even offering entire degree programs over the Internet. We can classify the electronic courses into *online* teleteaching sessions and *offline* computer-based training material (CBTs). Whereas many teachers have concentrated on the production of offline material, the hardware capabilities of modern PCs, workstations and transmission lines are now sufficient for transmitting lectures live over the Internet. Application software for online teleteaching is thus quickly gaining importance.

When a university professor plans to transmit lectures over the Internet, he/she first looks for software products supporting the specific needs. It turns out that specific teleteaching software cannot be found in the market. Thus the only possibly is to "abuse" existing software tools for teleteaching purposes. Typical examples include videoconferencing software, such as the MBone tools [13], Intel's ProShare™ or PictureTel™, or Web servers and Web browsers. In our own experiments we found that videoconferencing software and Web technology are far from being optimal for teleteaching. A detailed analysis of this situation is the topic of this paper.

In Section 2 we will discuss the requirements for a good teleteaching system (“what we need”). Section 3 describes the teleteaching systems we typically find at our universities today, and analyzes their deficiencies (“what we have”). Our own work on teleteaching tools at the University of Mannheim is described in Section 4. Section 5 concludes the paper.

## 2. What We Need – Requirements for TeleTeaching Tools

In order to analyze the requirements for teleteaching tools we classify the typical instructional settings we find at our universities into four *scenarios*. The scenarios vary in their degree of interactivity.

### 2.1. Remote Lecture Room (RLR)

In the *Remote Lecture Room* scenario two or more large lecture halls are interconnected with a high-speed network. During a lecture, three data streams are activated and transmitted from the instructor’s side: the video and audio stream of the instructor's camera and microphone, and the packet stream of the electronic whiteboard. Two channels are activated for each remote side in the reverse direction: the video and audio of the audience. During the lecture the instructor can see the students at the remote sites on his/her screen. Any questions they ask can be heard by everyone in all lecture halls. Students in all halls see a large-screen projection of the electronic whiteboard and all video streams. Similar equipment (hardware and software) is used in all lecture halls so that transmissions of lectures can take place from anywhere.

### 2.2. Remote Interactive Seminars (RIS)

In the *Remote Interactive Seminar* scenario small seminar rooms are equipped with a multimedia workstation and a high-speed network link. The transmission of the data streams and the projection are similar to those in the RLR scenario. Equipment cost is much lower since it is much easier to attain high-quality video and sound in small rooms. The RIS scenario bears a strong resemblance to that of a traditional videoconference.

### 2.3. Interactive Home Learning (IHL)

The *Interactive Home Learning* scenario connects computers of students in their private homes or student dorms via ISDN or modem to the network. They receive the audio, video and whiteboard streams from the teacher’s workstation via multicast. They can send their own audio and whiteboard streams to the group; video is also possible if the student has a camera. The main technical challenge in this scenario is the low bandwidth of the students’ links.

The IHL scenario can be effectively combined with the RLR or RIS scenarios: Students participate in a classroom lecture or seminar from their PCs at home. They can interact with the teacher and participate in discussions at any time.

### 2.4. Individual Learning with Computer-Based Training Modules (CBT)

The Computer-Based Training scenario is the one most widely used today. This scenario is asynchronous: A student accesses teachware on a server over the network. The teachware can include multimedia components such as video, audio and animations. The student works alone, at his/her own pace.

The four instructional scenarios are illustrated in Figure 1.

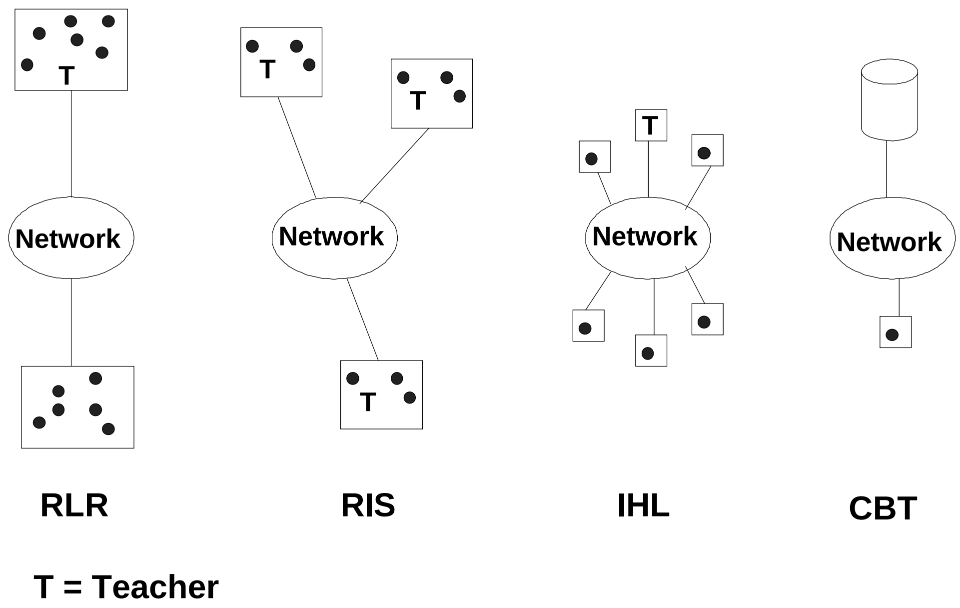


Figure 1: Four teleteaching scenarios

## 2.5. Requirements for TeleTeaching Tools

At the University of Mannheim we have been conducting teleteaching experiments with all four scenarios since 1995. For RLR and RIS our partners are the universities of Freiburg, Heidelberg and Karlsruhe. We found out very quickly that the videoconferencing and application sharing tools available in the market today were far from optimal for our purpose.

As mentioned above an online teleteaching session in the RLR, RIS and IHL scenarios consists of up to three data streams from each sender to the group of members:

- a digital video stream, showing the “”talking head” of the current speaker, or perhaps an experiment in the lecture hall,
- the audio stream of the current speaker, and
- a data stream produced by the electronic whiteboard. This data stream transmits the electronic document to be used in the session (typically a set of transparencies); during the session it also controls the turning of pages, transmits changes to the document itself or annotations by the teacher, etc.

Note that we consider document cameras as inappropriate technology in an all-digital environment. Nowadays the teachers prepare their material on a computer, data transmission is digital, and data presentation on the students’ machines is also digital. In our opinion it makes no sense to print out the transparencies, put them under a camera, use the analog lens system and low scan resolution of the television standards (525 or 625 vertical lines) to spoil the quality, then convert the camera output into a compressed digital video stream (with an additional loss in quality). It is much better to remain in the digital domain, load the transparency file directly into an electronic whiteboard, and then let the whiteboards of the session members interact via digital packet streams.

Thus, in our definition, a teachware system consists of client software and server software for video, audio and a whiteboard, and a network interconnecting them. From our experience we have derived a set of requirements for such a system which we present in the following.

### 2.5.1. Integrated User Interface

Teachers and students are confused by too many separate windows on the screen. They are also confused if they have to learn a different user interface for each scenario. Teleteaching and distance learning should not be restricted to computer experts; the user interface should be easy to understand and easy to operate. On the other hand the overall functionality required for the four scenarios is quite complex. As a solution we propose to have one integrated user interface, one interaction paradigm which can be easily *configured* to the specific needs of the scenario and of the teacher and/or student. Windows and control panels which are currently not of interest can be disabled.

For example an electronic lecture board might have the indexing and page-selection options disabled when used in the RLR scenario (i.e. in a classroom session). Or the floor-control panel might be disabled when used by a student at home in the CBT scenario.

### 2.5.2. Integrated Handling of Media

The possibility to use media other than text and still images are a key component of modern computer-based instruction. *Motion* and *interaction* really make the difference between paper-based teachware and computer-based teachware. It is thus crucial that teleteaching tools allow the *integrated use* of video clips, audio, animation and 3D models in the teachware. Microsoft's PowerPoint™ is moving into this direction. The controlled (synchronized) use of continuous media in a distributed fashion is still a research issue [12].

The importance of media types for teaching varies with the different fields. For example a professor of geography might want to show a video of a mudslide after explaining the physical factors contributing to the probability of such slides; a lecturer in anatomy might want to show a 3D model of a human head, rotating and decomposing it layer by layer while explaining the relative position of the organs; a professor of computer science might want to show an animation of network packets flowing through the Internet, illustrating multicast routing algorithms.

In any case, the teacher should not have to switch from the electronic whiteboard to other pieces of software, perhaps going back to the operating system shell, starting programs and manipulating complex control panels. Our experience shows that this is not accepted by the lecturers.

### 2.5.3. Collaboration Services

By "collaboration services" we mean an *electronic surrogate for the social protocols* in a classroom or group meeting. Social rules and protocols are the (often unconscious) basis for human interaction in most teaching and learning scenarios. Examples include raising hands, passing the right to speak or write on the board, pointing, and allowing people to join or leave the group. If people cooperate over a network an electronic replacement must be found for these protocols.

The most important collaboration service elements are

- Joining and leaving a session
- Raising hands (signaling)

- Granting and revoking the access to shared resources (“floors“), including the right to speak
- Pointing to shared objects
- Polling and voting
- Creating, joining, leaving and closing subgroups
- Side-talks (peer-to-peer or to subgroups).

We illustrate the importance of well-designed collaboration services with the example of traditional videoconferencing software used in an IHL scenario. The designers of videoconferencing software always assume that only a small number of conference rooms are interconnected (often just two). Participants can see and hear each other, and thus the traditional social protocols will work. But in the IHL scenario fifty students might participate in a lecture over the Internet. If every student created a camera stream to a video window on the screen of the teacher he/she would see fifty stamp-size talking head videos and not much else on the screen. Obviously it would be very hard to coordinate interaction with these students. An implementation of a floor control collaborative service could create a list of the remote students, with names or still images only, and raising hands could be shown by a red light. The underlying protocols and algorithms could enqueue interaction requests automatically, and when the teacher clicks on the list or a particular student this could automatically pass the floor (i.e. make the student’s machine the sender for the multicast session). Of course revoking the floor would also have to be implemented.

### 2.5.4. Synchronized Recording and Playback of Sessions

Once the audio, video and whiteboard annotation stream exist in digital format it is obviously desirable to record them on a disk for later playback. Students will then be able to review certain topics or complete lectures at any time. For optimal usage it is important that *all streams can be synchronized*, decomposed into small units corresponding to single topics, and accessed via an index of keywords.

In order to explain the importance of synchronization let us look at an example from a computer science lecture. The professor is planning to explain an algorithm where data packets flow over a graph representing a network. In preparation of his lecture he draws the topology of the graph, with nodes and edges; during the lecture, while interacting with his students, he annotates the graph with colored arrows representing the packet flow (see Figure 2). The dynamics of the algorithm are explained by annotating the graph. In other words, without reproducing the dynamics of the annotations in the recording, essential information is lost.

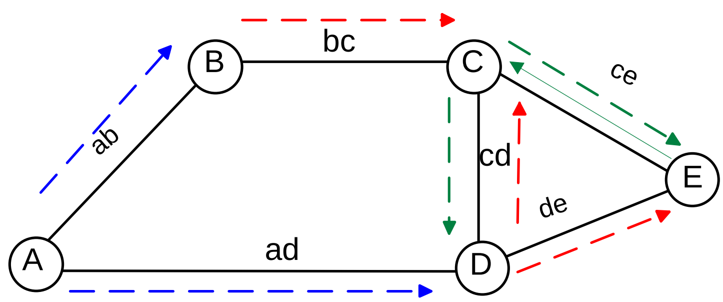


Figure 2: An example for the use of dynamic annotation in the classroom

### 2.5.5. Multimedia Library Server

We expect a large increase in the amount of multimedia teachware in the future. As we know from classical database technology (for storing formatted data) it is crucial for the availability of shared data that central repositories are established, with efficient storage structures optimized for the different media types, and with indexing and retrieval facilities. In the long run it is desirable to be able to find relevant teachware anywhere in the world, to retrieve it over the network, and to use it locally for teaching and learning.

The only widely accepted storage and retrieval formats are those of the World Wide Web. The markup language of choice is html, and the media types and formats are the ones for which plug-ins exist for the most popular Web browsers. Media examples include GIF images, Java applets, VRML models and MPEG-1 or AVI video clips. In this context the teachware server is a Web server, and the teachware client is a Web browser.

Whereas this approach might be acceptable for the CBT scenario it is certainly not appropriate for online teaching. The main reasons are

- There are no standardized streaming protocols for continuous media on the Web. A new Internet protocol called RTSP is under development for this purpose, but not yet widely implemented [16].
- Web browsers do not support online group interaction
- Web browsers do not support annotations
- The Web protocols do not run over multicast IP; data distribution to groups is very inefficient.

### 2.5.6. Annotations

Note-taking and annotations in books and in course materials are a cornerstone in traditional learning in higher education. Annotations “personalize“ the teachware, they make learning more effective. Unfortunately annotations on electronic material are not very widely used today; they are either difficult to do or not even supported at all. We are convinced that this is the major reason why most people still tend to print before reading.

In addition to “static“ annotations of a document the stepwise production of a complete drawing or text with *dynamic annotation* is a didactically powerful way of explaining the behavior of a system in many disciplines. For example a teacher can bring half-completed graphics to the classroom and develop the missing pieces interactively with the students. From a didactic perspective dynamic annotations find their place somewhere between static annotations and animation.

An evaluation of our own electronic teachware has shown that student attention is much higher during phases of “dynamic annotation” than during phases with static material. As explained above dynamic annotation requires the system to store not only the final version of the document but also the sequence of production of the annotations over time.

### 2.5.7. Network Service with Multicast and QoS

Since we must transmit the continuous media streams for digital video and audio along with the discrete media streams for the electronic whiteboard we need an underlying network that provides *quality-of-service* (QoS) guarantees. During the last decade much research was done on

QoS guarantees in packet-switched networks, and it is now well understood that bounds on delay, delay jitter and loss rate are the most relevant requirements. And of course the network must provide guaranteed data rates for the (compressed) media streams.

*Multicast support* within the network is another important requirement. The RLR, RIS and IHL scenarios are all group scenarios where a (possibly changing) sender transmits data packets to a group of receivers. If a network only supports point-to-point links then multipeer transmissions are very inefficient; in fact they might be unfeasible for large groups. This is illustrated in Figure 3. In a network with integrated multicast technology the internal nodes are able to understand group addresses, set up tree routes to multiple receivers, and duplicate data packets as far ”down the tree” as possible. Hosts are able to join and leave multicast groups at any time. The Multicast IP protocol of the Internet, used in the MBone overlay network, is an example.

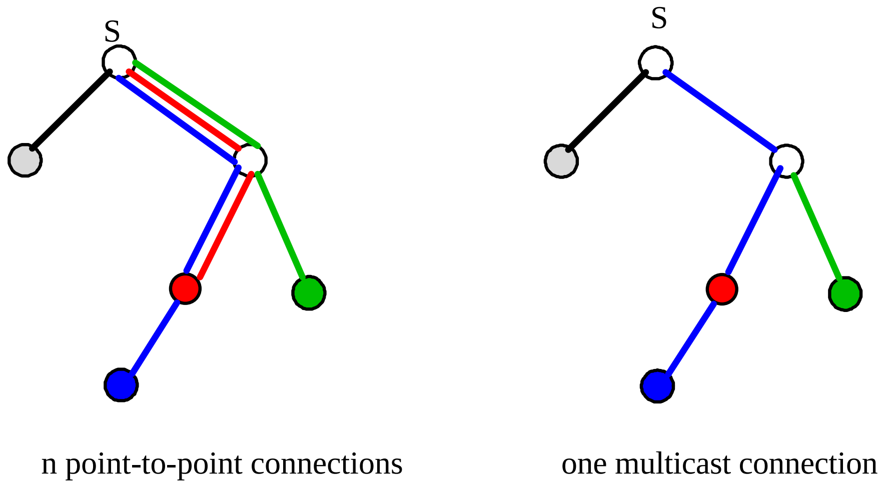


Figure 3: Multiple point-to-point connections vs. one multicast connection

In Section 3 we will see that neither the current Internet nor native ATM or ISDN services fulfill both requirements (QoS and multicast).

### 2.5.8. Access Management and Security

As long as access to universities is free and all teachware is in the public domain, there is little need to protect and administrate the access to the online sessions and to the teachware server. On the other hand, the development of multimedia teachware is very expensive; estimates are that between 100 and 300 hours of work are needed to produce multimedia teachware for a one-hour session. Therefore more and more universities are restricting access to their materials, and students must pay tuition fees if the wish to participate in multimedia courses. As a consequence session data in transit must be secured against stealing, and access to the multimedia library server must be protected.

The basis for all security is encryption. Whereas the protection of discrete media against intruders is well understood *block ciphers for continuous media* are a subject of current research: they must be very efficient because transmission is in real-time and high volumes of data are involved, but they should not sacrifice security. Crypto-algorithms for group communication are not very well understood either. And since group members can join and leave a teleteaching session at any time the key distribution problem is difficult to solve. In some cases multimedia groupware products coming from the United States have their security features disabled because exporting powerful encryption technology is illegal. A good teleteaching software package should have powerful yet efficient crypto-algorithms built in.

If access to the online sessions and to the teachware server must be limited administration can become a problem. Management tools must be set up to handle lists of teachers, students and courseware modules, and to assign individual access rights. Access management systems for electronic teachware are beyond the scope of this paper.

## 3. What We Have – An Analysis of TeleTeaching Tools in Use Today

We will now take a quick look at the software tools and network infrastructure available today and compare their capabilities with the requirements derived in the previous section.

### 3.1. Application Software

Most teleteaching projects at universities are either based in videoconferencing/CSCW tools or on WWW technology.

#### 3.1.1. Videoconferencing/CSCW Tools

Systems for videoconferencing and CSCW can be classified into *collaboration-unaware* and *collaboration-aware*. The former are based on a technique called *application sharing*. The principle of application sharing is to insert an additional software layer between the window system and the application program. All the events coming from the application program are intercepted and multicast to the remote window systems participating in the session. For example a function call placing a button on the screen is such an event; the data packet created by this event on the local system is copied and transmitted to all remote sites. Events from all the participating window systems are serialized and enqueued for handling by the application program. Since application programmers rely on a standardized set of events when writing interactive (window-based) programs it is possible to write a universal application-sharing layer.

The advantage of application sharing (or the collaboration-unaware approach) is that arbitrary applications can be shared with remote partners; application programs need not be modified. The effect is always WYSIWIS (What You See Is What I See). Whatever functions an application provides (e.g., annotations) is available on all systems of the group. Data is only held in one place, where the application program is running. Disadvantages are that collaboration services are (by definition) not part of the applications, and must be provided as add-ons. Also, as a consequence of the strict WYSIWIS principle it is not possible to form sub-groups or support private annotations. And all participants must have the same window system; otherwise the transmission of window interface events would not work. Many commercial products such as ProShare™, PictureTel™ or NetMeeting™ fall into the application sharing class.

Application sharing tools are quite weak in most of the requirements we have listed in Section 2.5. Media integration is almost unfeasible because sharing video or audio via window events is impossible. Integration of strong collaboration services is impossible by definition. And these tools typically run over native ISDN or TCP/IP, point-to-point protocols which are unable to communicate to large groups efficiently.

Collaboration-aware systems are designed as distributed systems; the software is written with group interaction in mind. They can implement more sophisticated sharing and communication mechanisms.. Collaboration services such as session control, floor control, telepointers and annotations can be supported in an integrated fashion.

Each site has a local copy of the data. The number of packets flowing over the network is usually much lower, the response time is faster. Local annotations can be supported as well as side-talks and sub-groups. Heterogeneous systems can participate in a session as long as the “speak“ the protocol of the tool. However it is not possible to share arbitrary applications, and end users must learn a new interface. The types of files that can be imported into the tool is limited. Shared whiteboards such as the MBone tool wb (see [13]) are a typical example for this class; more general collaboration environments were developed in the JVTOS project [2] and in the MMC project [14].

Many CSCW tools of both classes were developed over the last two decades but unfortunately they have not converged to one generally accepted solution. And the control protocols for collaborative services are not standardized so that a user of one system cannot cooperate with the user of another system. Only very recently ITU-T has developed a new family of standards for teleconferencing (T.120 [10]), so there is hope for better interoperability in the future.

### 3.1.2. Web Technology

Other teleteaching projects rely heavily on the technology of the World Wide Web. They often start with the CBT scenario, develop material for the Web and then try to use it in online sessions. The major strength of this approach is the possibility to easily include educational material from all over the world; there are millions of authors writing for the Web. The file types are very well understood and very widely used, not only the html markup language but also GIF, AVI, MPEG, VRML, etc. Media integration is a strong point of the Web. And powerful authoring tools are available, enabling more and more teachers to produce multimedia teachware.

The main drawback of this approach is that neither the Web browsers nor the Web servers were designed with group communication in mind. Add-ons must be written in order to jointly view a Web page, annotate, manage the group, turn pages, etc. The communication protocol of the Web is http which runs over TCP; it is inherently point-to-point, not multicast, and large groups cannot be supported efficiently. In their pure form Web browsers and Web servers are only appropriate for the CBT scenario; some research projects are attempting to extend their functionality to online teaching. We claim that the use of html and media types and formats of the Web is a good idea, but the use of traditional Web servers and browsers for the RLR, RIS and IHL scenarios is not.

## 3.2. Authoring on the Fly

Whereas very powerful software is available for video editing on the PC, the recording and editing of whiteboard annotations over time (dynamic annotations) is usually not possible, and playback in synchrony with audio and video is not supported by commercial products. Professor Ottmann and his research group in Freiburg have developed an interesting prototype system called “Authoring on the Fly” (AOF) [1]. Dynamic annotations are time-stamped, recorded, and synchronized with the audio stream. The set of AOF tools includes a recording whiteboard for the teacher and a viewer for the students, allowing them to playback the material in a synchronized fashion. Lectures can be decomposed into logical units, and the viewer supports an easy-to-use visible scrolling.

## 3.3. Network Technology

When we analyze the network access available to teachers and students we find that almost everyone is connected to the Internet. Some also have access to ISDN; very few institutes have direct access to an ATM network.

### 3.3.1. ISDN

ISDN is typically used as a replacement for plain old telephone service. If telecollaboration traffic exists over ISDN, it is based on proprietary PC solutions, such as Intel's ProShare™ or the PictureTel™ system. Only very few teachers or students have facilities to route Internet traffic over ISDN.

An advantage of ISDN is the fact that it is circuit-switched; Quality-of-Service parameters are always guaranteed, and several ISDN B-channels can be combined for higher bandwidth. Long-distance bandwidth is typically 64 kbps (one channel), 128 kbps (two channels) or 384 kbps (6 channels). But a major disadvantage for our purpose is the *lack of support for multicast* in the ISDN network: it only provides point-to-point links. Also, the cost of long-lasting long-distance ISDN connections is still very high in Europe.

### 3.3.2. ATM

ATM is a technically very interesting solution for multimedia traffic: It is connection-oriented, supports QoS in its traffic models, and offers very high bandwidths. There is a small number of TeleTeaching trials in Europe over *native* ATM, e.g. between ETH Zürich and EPFL Lausanne, and between TU München and the University of Erlangen in Germany. In these trials ATM virtual circuits are set up between the participating locations, and special-purpose audio equipment and ATM cameras are connected directly to the ATM switches on both ends.

In principle ATM supports multicast, but in practice it is often difficult to set up multicast circuits if the ATM switches come from different vendors. And the number persons reachable with native ATM is still very small.

### 3.3.3. The Internet Protocol Stack

The Internet protocol stack can also include ISDN links and ATM virtual circuits in sub-networks, but it always adds the IP layer on top, and that guarantees universal connectivity between all end systems. Also all modern IP implementations on workstations and PCs support the multicast protocols of the Internet (multicast IP), and as explained above, multicast is crucial for our typical point-to-multipoint transmission scenarios.

A major disadvantage of the current Internet protocol architecture is the *lack of QoS support*, i.e. all connections are best-effort. This is also true for multicast IP and the MBone; it can lead to very bad audio and video quality. Providing multicast *and* QoS guarantees is one of the major challenges for the next-generation Internet protocols.

# 4. TeleTeaching Tools Developed at the University of Mannheim

## 4.1. dlb - The Digital Lecture Board

Our analyses in Sections 2.5 and 3 indicate that standard video conferencing solutions are far from optimal for computer-based distance education; they have not been designed for this purpose. This is especially true for the shared whiteboard. In order to overcome the weaknesses of existing whiteboards (such as the MBone whiteboard wb), we have decided to develop our own *digital lecture board* dlb. It is an extended whiteboard tailored to the specific needs of teleteaching.

We have implemented a prototype of the dlb fulfilling many of the requirements mentioned in Section 2.5. Since the source code for the MBone whiteboard wb is not publicly available we had to write our own code from scratch. To allow for a high degree of portability, we implemented the prototype in C++ and the Tcl/Tk scripting language [[15] [17], and we took great care to reuse only components available on all major hardware/software platforms (e.g. ghostscript for rendering postscript pages). The current version runs on the Unix systems IRIX (SGI), Solaris (SUN), and Linux (PC). The following features are supported:

- reliable multicast/unicast transmission using smp (scaleable multicast protocol)
- late join (students joining the session too late receive all materials up to date)
- media formats: postscript, gif, ppm, ASCII
- graphical objects: freehand lines, lines, arrows, polylines, polygons, rectangles, ovals
- editable text objects with different font types, styles, and sizes
- joint editing of graphical objects and text (source synchronized)
- clipboard functions: cut, copy, paste, undo, group, ungroup
- zooming
- on-line and off-line mode (provides a private workspace)
- SGML-like document format (allows storage and retrieval of dlb documents)
- partly configurable, easy-to-operate user interface
- collaboration services (floor control, session control)
- telepointers
- RTP-compliant network protocol, compatible with the AOF whiteboard [1].

Figure 4 shows a screen shot of the dlb. The user interface looks like common Windows applications. Similar to graphic or text processing software, the dlb provides text, graphic and editing functions located below the menu bar. The page selecting utility on the upper left provides users with access to the dlb online document. Pages can be created or accessed by mouse click. The *Live* button is used to leave the online session temporarily: the user can then create pages or graphical objects without transmitting them to the other participants. As soon as he/she toggles the *Live* button again, the prepared material can be transmitted to the group. The *Annotation* button enables users to attach private annotations to online documents. The annotations are only stored in the offline dlb-document. The TP button activates a telepointer. Currently, we are supporting up to three simultaneous telepointers with different colors. Below the page selection panel the dlb provides information about the current participants in a session. By clicking on a participants' name the user can get more information on the person. The information is based on RTCP control messages which are exchanged periodically between the distributed dlb instances.

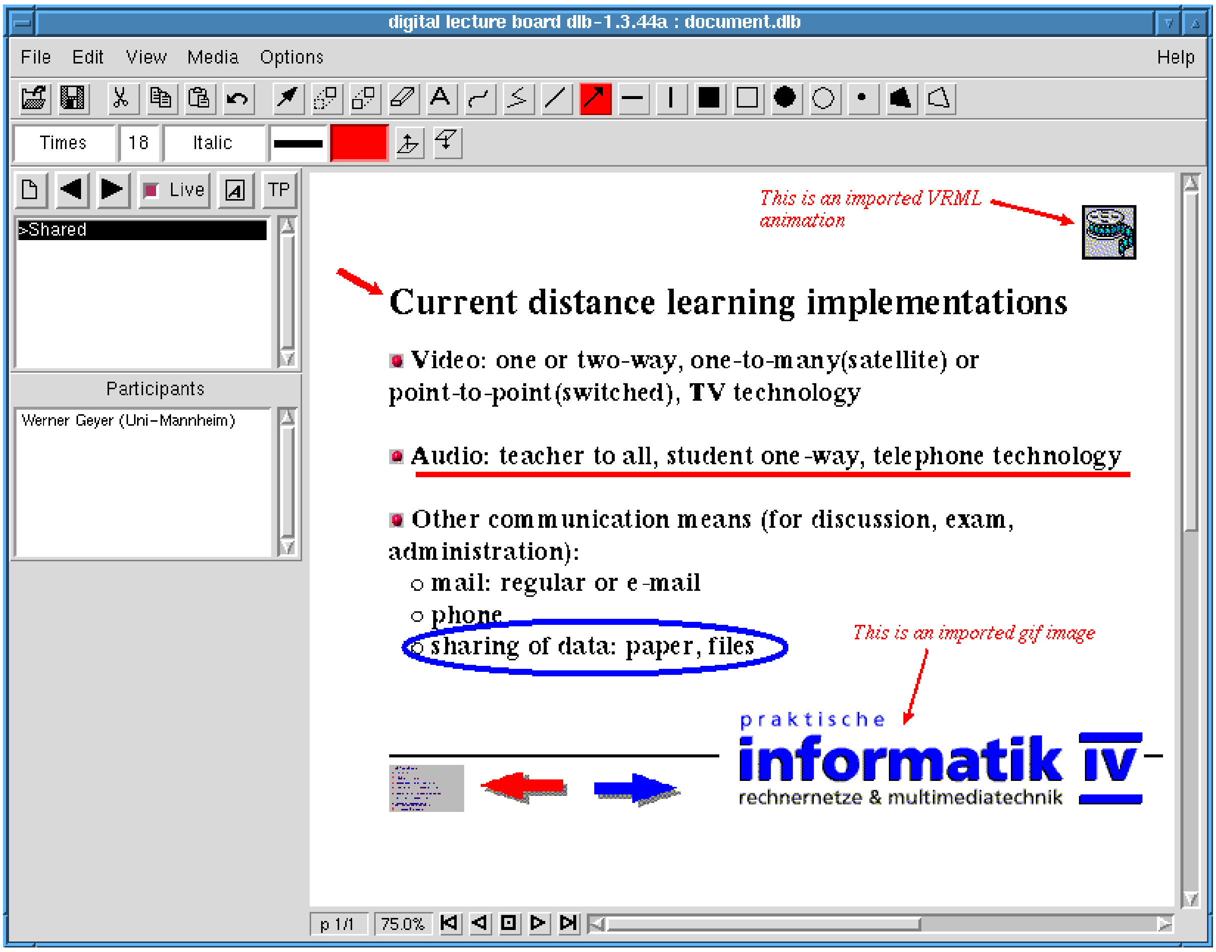


Figure 4: User interface of the digital lecture board

### 4.1.1. System Architecture of the dlb

The dlb system comprises several components as indicated in Figure 5. On the *application level*, the core part of the dlb embeds functional modules which are implemented as encapsulated C++ classes. The postscript module (*ps*), for instance, supports rendering of postscript pages by interfacing to the ghostscript interpreter (*gs*). The telepointer module (*tp*) provides a distributed, shared pointing device. The *mdb client* and *vcr client* allow for remote access to a multimedia database (MDB) and the MBone VCR, respectively. Additional modules are floor control (*fc*) and session control (*sc*), interfacing to the collaborative services model (*csm*). New modules can be implemented to extend dlb's media capabilities with new media types such as html, jpeg, animations, etc.

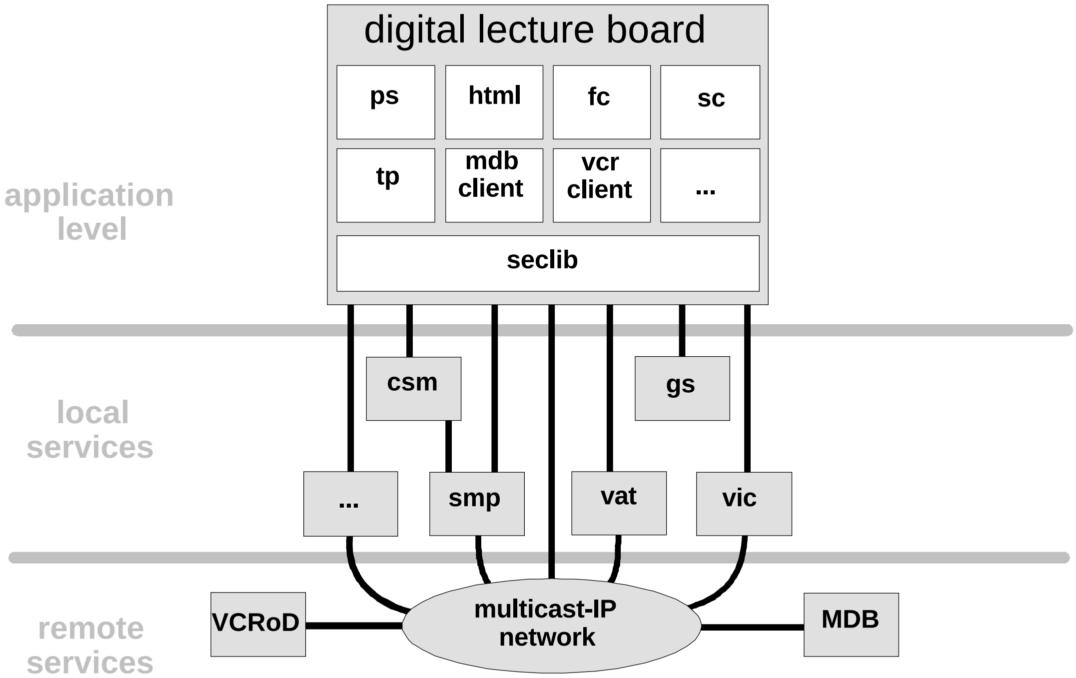


Figure 5: Components of the dlb system architecture.

Besides embedded modules, the dlb makes use of *local services* such as ghostscript (*gs*) for rendering postscript pages, *vat* for audio communication, *vic* for video communication, the collaborative services model (*csm*) for managing collaborative services, and the scaleable multicast protocol (*smp*) for reliable multicast transmission. Remote services accessed via a network are, for instance, the Video Conference Recording on Demand service (the MBone VCR [8]) or a multimedia database (*MDB*). Note that vic, vat and gs are existing software components whereas csm and smp were specifically developed for the dlb. Instead of integrating these components into the dlb core, we implemented them as separated service applications which can be accessed by a well-defined application programming interface:

- **scaleable multicast protocol** (smp): When the dlb project began no commonly accepted reliable multicast protocol was available on the Internet. Most protocols are either heavy-weight, or they are integrated in applications [4]. So we decided to implement our own scaleable multicast protocol smp which is based on the SRM paradigm (Scaleable Reliable Multicast) of wb but it is enhanced with a local scoping mechanism for even better scalability. In contrast to SRM, smp is implemented in a separate process which provides reliable multicast transmission services to arbitrary applications. Additional features are source ordering, late join, and rate control (e.g. for low bandwidth links). An application using smp does not have to be concerned with reliability issues.
- **collaborative services model** (csm): As indicated in Section 2.5, collaboration services provide an electronic surrogate to compensate for the lack of inter-personal communication. The csm implements enhanced floor control and session control mechanisms and policies. Floor control realizes concurrency control for interactive, synchronous cooperation between people by using the metaphor of a *floor*. A floor is basically a temporary permission to access and manipulate resources (e.g. a shared drawing area). Session control denotes the administration

of multiple sessions with its participants and media. Session control increases social awareness in distributed work groups because members have knowledge of each other and their status in the session. The csm keeps the collaboration state (the relationships between participants, floors, resources, and sessions) in a single object-oriented model. The model is replicated on each participant's workstation and held consistent by using an optimistic synchronization scheme. Like smp, csm is implemented in a separate process which provides services to arbitrary applications. This is specifically useful if several applications are involved in the same session. For a more detailed description of the csm component see [7].

### 4.1.2. Security Issues

The digital lecture board also includes a user-oriented security concept [6]. We provide three predefined profiles - *innovative companies*, *financial services* and *public research* - which include encryption algorithms like CAST, DESX and IDEA. The concept is integrated directly into the core part of the dlb as an additional security layer (see Figure y and z). We have implemented a security library providing full compatibility with the Open-PGP standard, i.e. dlb‘s RTP data packets are wrapped in OPGP packets. We then use either unreliable UDP services or reliable smp connections to transmit the OPGP/RTP packets. The security functionality can be accessed and controlled through dlb's graphical user interface and via command line parameters.

## 4.2. Remote Control for Java Animations

Animations are a powerful means to illustrate the dynamic behavior of a system. With the proliferation of Java-enabled browsers and the popularity of the Java language more and more animations are becoming available. The Java system allows a user to download a Java applet and run it locally on his/her computer. All interactions of the user with the applet happen on the local system.

In order to use applets in teleteaching a remote control mechanism is required. Consider the RLR or RIS scenarios and a teacher who wants to use a Java animation. He/she will load the Java program on the machine in the lecture hall or seminar room; but how does a copy get loaded on the remote machines? And how are interactions of the teacher transmitted to the remote machines so that all students see the effects in synchrony (and in particular, synchronous to the audio)?

The Java Remote Control tool developed at the universities of Mannheim and Heidelberg solves these problems [12]. The Java animation code is extended slightly with a "distributed-control-object" class, talking to a central server. All events created by the local operator are transmitted to the server, broadcast to all other copies of the Java program, and acknowledged. Only then the effect of the event takes place at all sites. This is illustrated in Figure 6. A mechanism for setting synchronization points is also provided. In this way there is no need to broadcast entire pixel screens to remote locations (as in window sharing systems, such as Shared X); the graphics are updated on each machine by its local copy of the Java code, and synchronous appearance is still guaranteed. The Java remote control tool is implemented and fully operational.

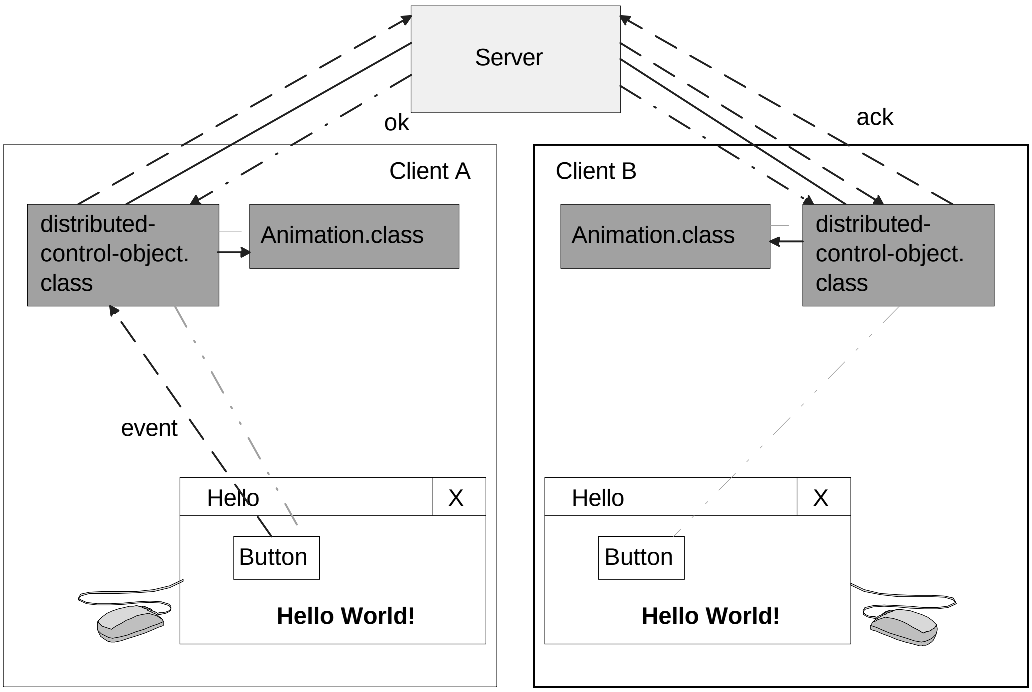


Figure 6: Flow of control packets with the Java Remote Control tool

### 4.3. MBone VCR - Recording and Playback of MBone Sessions

The University of Mannheim, in cooperation with ICSI in Berkeley, has designed and implemented an MBone Video Conference Recorder (MBone VCR). It provides functions to record and play back Mbone sessions with multiple multicast multimedia data streams from different applications. A recorded session can consist of as many multicast channels as the user wants to record. The channels can originate from different locations and applications [8] [9].

During recording the MBone VCR synchronizes the multimedia data streams, based on information provided by RTP, the Real Time Protocol of the Internet. To play back the data streams, the MBone VCR transmits the recorded data, using the original timing and packet format. In addition to the VCR itself he receivers have to run the audio/video applications such as vic or vat to watch and listen to the recorded data.

The MBone VCR provides additional interesting features, such as indexing of the recorded data. The user can easily skip certain phases, for example, a certain speaker, of a session. Indices allow the user to jump back and forth between marked scenes. Other features, such as fast forward and rewind, are similar to those of a home VCR. Sessions to be recorded can be selected from the MBone Session Directory (sdr) which provides the user with lists of current and upcoming MBone events.

The most recent version is called MBone VCR on Demand (MvoD, [8]). It offers a solution for interactive *remote* recording an playback of multicast video conferences. A user can interactively record audio/video conferences on a remote server, controlling it with a local client application, and later the same user or even other users can play the session back on demand per multicast or unicast. The MVoD architecture consists of three components:

- the *MVoD Server* deals with the user management and the session management,
- the *MVoD Client* offers a graphical user-interface to access the MVoD Service, and
- the *RTP DataPump* which is responsible for the recording and playback, the synchronization and the administration of the RTP data streams.

For the communication between the above components, a number of protocols were developed:

- the VCR Announcement Protocol (VCRAP) with which the server announces its service to the clients,
- the VCR Service Access Protocol (VCRSAP) through which the clients have access to the server,
- the VCR Stream Control Protocol (VCRSCP) that the clients use to access and controll a session on the server, and
- the RTP DataPump Control Protocol (RDCP) that the server uses to control the RTP DataPumps (one per session).

Moreover, an interface to the Session Announcement Protocol (SAP) was implemented through which the MVoD Server learns about ongoing MBone sessions. A widely use tool that implements SAP to announce MBone sessions is sdr (the Session Directory). For the real-time data transfer the RTP DataPump fully implements RTPv2. The overall architecture of MVoD is show in Figure 7.

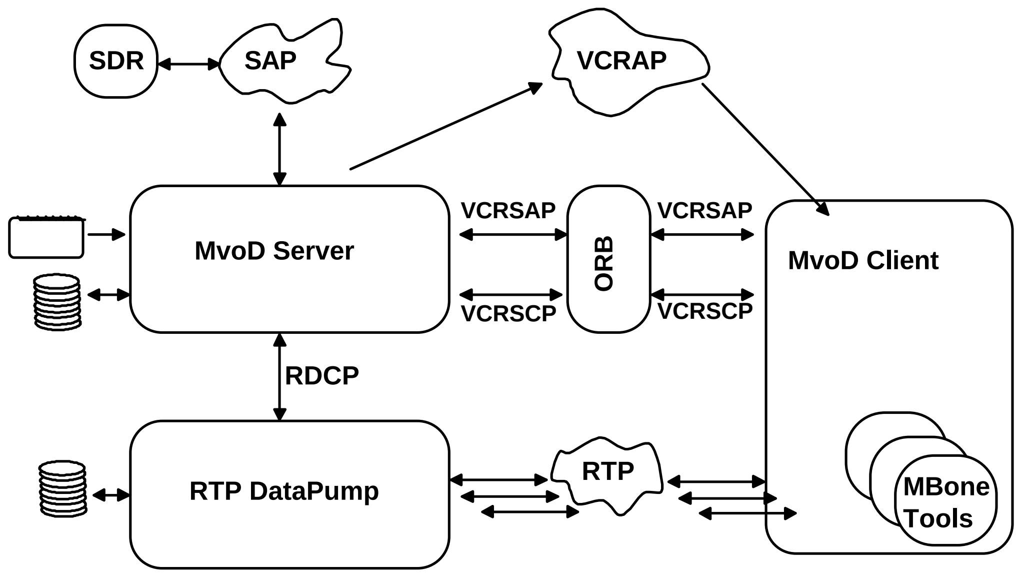


Figure 7: Architecture of the MBone Video on Demand system

## 5. Conclusions and Outlook

We have classified teleteaching activities over the Internet into the four scenarios RLR, RIS, IHL and CBT. For these scenarios we have derived a set of technical requirements for good teleteaching tools. We claim that neither the existing videoconferencing/CSCW systems nor Web technology provides the desirable functions.

We have presented the teleteaching tools under development at the University of Mannheim: the digital lecture board, the Java Remote Control tool, and the MBone VCR. If used in conjunction with the popular MBone tools vic, vat and sdr, these tools provide a very rich and powerful teleteaching environment. Not all the functions of our list in Section 2.5 are implemented yet but we are working on them.

Some presenters seem to believe that multimedia teachware is using PowerPoint on a laptop, and explaining everything in the world with bullet lists. We claim that the real potential of multimedia teachware comes with the use of *active components*, such as audio, video, dynamic annotation, animations, simulations and 3D models. And it must be possible to use all of these in a distributed environment. In spite of the multimedia hype in the public media this is not possible with the commercial products we can buy today; neither with the software tools nor with the network technology.

For the future we expect *really innovative* teaching and learning approaches to appear that go beyond a "distributed simulation" of traditional teaching and learning scenarios. For example a worldwide, interactive, distributed stock exchange game could introduce students in business administration to the mechanisms of international stock markets. Or a online teacher evaluation tool could allow students in the IHL scenario to give an immediate feedback to their teacher: they would have sliders for "too easy – too difficult", "too fast – too slow", etc. on their screens, and the teacher would see a number of small gauges on his display, averaging the students' votes.

More research, development and experimentation is needed to make good teleteaching with multimedia a reality.

### References

[1] Ch. Bacher, Th. Ottmann: Tools and Services for Authoring on the Fly. Proc. ED-MEDIA 1996, Boston, available from AACE on CD-ROM, http://www.aace.org

[2] G. Dermler et al.: Constructing a Distributed Multimedia Joint Viewing and Tele-Operation Service for Heterogeneous Workstation Environments. In: IEEE Computer Society - Technical Committee on Distributed Proceedings (ed.): Proceedings of the Fourth Workshop on Future Trends of Distributed Computing Systems, September 22 - 24, Lisbon, IEEE Computer Society Press, Los Alamitos 1993, pp. 8 – 15

[3] W. Effelsberg, R. Steinmetz: Video Compression Techniques. dpunkt.Verlag, Heidelberg, 1998

[4] S. Floyd, V. Jacobson, C. Liu, S. McCanne, L Zhang: A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing. Proc. SIGCOMM 95, August 1995, pp. 342-356

[5] W. Geyer, W. Effelsberg: The Digital Lecture Board – A Teaching and Learning Tool for Remote Instruction in Higher Education. Proc. ED-MEDIA 1998, Freiburg, available from AACE on CD-ROM, http://www.aace.org, pp. 63-69

[6] W. Geyer, R. Weis: A Secure, Accountable, and Collaborative Whiteboard. Proc. Interactive Distributed Multimedia Systems 1998 (IDMS 98), Oslo, to appear

[7] V. Hilt, W. Geyer: A Model for Collaborative Services in Distributed Learning Environments. Proc. Interactive Distributed Multimedia Systems 1997, Darmstadt, Springer LNCS 1309, 1997, pp. 364 -375

[8] W. Holfelder: Interactive Remote Recording and Playback of Multicast Videoconferences. Proc. Interactive Distributed Multimedia Systems 97 (IDMS 97), Darmstadt, Springer LNCS 1309, 1997, pp. 450-463

[9] W. Holfelder: Aufzeichnung und Wiedergabe von Internet-Videokonferenzen. Dissertation, Shaker-Verlag, 1998 (in German)

[10] ITU-T: The T.120 Standard. All documents available directly from ITU's bookstore. See http://www.itu.ch/itudoc/itu-t

[11] Ch. Kuhmünch: A Multicast Gateway for Dial-In Lines. Submitted for publication. Available from http://www.informatik.uni-mannheim.de/informatik/pi4/publications/

[12] Ch. Kuhmünch, Th. Fuhrmann, G. Schöppe: Java Teachware – The Java Remote Control Tool and ist Applications. Proc. ED-MEDIA 1998, Freiburg, available from AACE on CD-ROM, http://www.aace.org, pp. 70-75

[13] V. Kumar: MBone – Interactive Multimedia on the Internet. New Riders Publishing, Indianapolis, 1996

[14] Multimedia Collaboration Homepage. Url: http://www.prz.tu-berlin.de/docs/html/MMC, 1996.

[15] J.K. Ousterhout: Tcl and the Tk Toolkit. Addison-Wesley, 1994

[16] Real-Time Streaming Protocol RTSP, Internet RFC 2326, ftp://ftp.isi.edu/in-notes/rfc2326.txt

[17] B.B. Welch: Practical Programming in Tcl and Tk. 2nd edition, Prentice Hall, New Jersey, 1997