A Systematic Approach to the Automatic Conversion of a „Live" Lecture into a Multimedia CBT Course

Claudia Schremmer, Volker Hilt
Lehrstuhl für Praktische Informatik IV

University of Mannheim, Germany

{schremmer,hilt}@informatik.uni-mannheim.de

Abstract. *In this paper we present a procedure and a software system to automatically create a CBT version from the transparencies and audio/video recording of a classroom lecture. Whereas more and more professors put their transparencies and exercises on the Web, the tremendous amount of extra work required by traditional CBT authoring prevents them from producing a multimedia version for self-paced learning. Our conversion system bridges this gap: we decompose a set of transparencies created with an arbitrary text editor into a linked list of HTML documents and automatically add the appropriate audio and video tracks. Supplementary material such as animations, simulations, exercises and self-test questionnaires can be integrated based on a configuration file.*

KEYWORDS. COMPUTER-BASED DISTANCE LEARNING, COMPUTER-BASED TRAINING, INSTRUCTIONAL DESIGN, AUTHORING TOOL, AUTHORING PROCESS, MULTIMEDIA COURSE

INTRODUCTION

In 1995, the University of Mannheim and the University of Heidelberg initiated a joint teleteaching project [Eff97, EGE97] with the aim to export lectures in physics from Heidelberg to Mannheim and lectures in computer science from Mannheim to Heidelberg. In this project, lecture rooms and seminar rooms have been equipped with multimedia technology (i.e. multimedia-workstations, connections to an ATM Network, microphones, loudspeakers, video cameras and high-quality video beamers). The software employed for the transmission of lectures and seminars are the Internet's MBone tools *vic* (video conferencing tool), *vat* (visual audio tool) and *wb* (whiteboard).

Complementary to the transmission of lectures and seminars, our group at the University of Mannheim is developing material for *asynchronous* learning [HK99]. In this scenario, students are provided with course material for lectures and seminars. The course material comprises not only the lecturer's transparencies, but also additional information such as recorded audio and video, in-depth text on selected items, animations and exercises that interactively return feedback and a sample solution to the students.

In order to record the audio and video data streams of a lecture, we are using a tool for remote recording and playback of multicast videoconferences called *MBone Video Conference Recorder* or VCR [Hol97]. During the Summer semesters 1998 and 1999 we have recorded audio and video of the course *Multimedia Technology* with this tool and during the Winter semester 1998/99 we have recorded the course *Computer Networks*. Based on the audio and video material and the transparencies of the lectures we have designed and implemented a system to automatically create a CBT version of the courses.

In this paper we present our approach to the automatic conversion from one single *primary format* of the documents into the above-mentioned full CBT. The following section gives a brief overview of related work. Section Three discusses the objectives of asynchronous learning scenarios and derives the architecture of our CBT. Section Four details the linking of active elements to the transparencies. An *Instructional Design* is presented that results from psychological research on learning-encouraging environments. Section Five details our experience. The paper concludes with an outlook.

RELATED WORK

Many developments arise from the current intensive research in the domain of asynchronous computer-based learning. We can only cite a few of the arising online-systems.

The University of Distance Learning Hagen is predestinated for development of distance training material [Hag99]. As they also have to manage admission and access control to online-material, the focus of the University of Hagen is set on classroom management. The Technical University of Dresden and TU Bergakademie Freiberg have developed *a Java-based Teleteaching Kit* [Dre98]. The project's objective was to develop and evaluate modular internet-based software packages for teaching material. A *Digital interactive Lesson in Students' Training* is elaborated by the University of Berlin [Ber99]. Both approaches rely on learning environments that have been developed within the projects. In contrast to the approach presented in this paper, the effort to produce material within these environments is enormous. The TU Darmstadt has recently released its generic, networked, multimedia learning environment [Dar99]. Depending on a learner's profile, contents of his personalized learning environment are created individually.

The University of Freiburg pursues a goal similar to our's: based on live lectures, research in Freiburg evolved in the direction of recording not only the audio, but also the whiteboard streams and the annotations on the whiteboard. This approach is called *Authoring on the Fly* [BMO97, BMOW97]. Differences to our CBT material are mainly due to a different technique. We intend to combine both developments within a short time.

ASYNCHRONOUS LEARNING

Objectives

The traditional method of teaching „face-to-face” bears two obvious constraints: it is *time* and *location* dependent. While teleteaching scenarios afford independence of location, independence of time and independence of learning speed can be best achieved by asynchronous scenarios. One of the main goals of computer-based learning material thus is to deliver educational material to students who can process it selectively and of their own pace depending on their learning preferences. In our university context, such material usually accompanies a lecture held in the synchronous scenario.

In contrast to synchronous learning scenarios, in which transfer of knowledge and capabilities to students often is rather „passive“, learners in asynchronous scenarios are „active“. Due to this fundamental conceptual difference, the amount of material presented may exceed that of the lecture, and its level of detail may vary more. While synchronous teaching often is designed to give a general overview of a certain aspect and consequently does not enter the depths of topics, asynchronous material is designed for in-depth use. Students use the material interactively; this permits them to skip features already known or not relevant to them. On the other hand, easy and especially interested learners might delve deeper than others. Examples for the realization of interactive components comprise full text, animations, simulations and exercises.

Time independence denotes that students are able to learn whenever it meets their needs, at their personal speed. To achieve true time independence, a student should be able to attend a lecture independent of the time when it was originally given. A classroom lecture is an important example of instructional material that should be accessible asynchronously at a later time. To be widely accepted, a good asynchronous scenario must support location independence. Thus, material has to be stored on a teachware server accessible through the Internet.

Our approach to the automatic conversion of a live lecture into a multimedia CBT rests upon the fact that most course material already exists in digital form. With a minimal amount of human resources audio and video is recorded, digital transparencies are transformed into a more appealing presentation format, and a CBT evolves.

Design Requirements

Most lectures at universities already exist, in other words, slides used during the lecture have been created some time ago. To keep the authoring effort at an acceptable level, the existing material is updated only for the needs of a current lecture. Every lecturer has his own preferences for the text processing software to create the slides. We claim that letting everyone use his own favorite tool is a major advantage of our system: authors will be more efficient, and acceptance in practice will be better.

Which are the precise requirements? A lecturer has prepared his course with his text processing tool, say MS Word, Framemaker or Latex. Thus, all documents are in the *primary format* of the authoring tool [HK99]. We claim that several different *presentation formats* for different learning/teaching scenarios should be produced from the same primary format.

Let us consider the required format for the synchronous teleteaching scenario. As the lecture is transmitted via the MBone tools wb, vic and vat and as the whiteboard wb only processes *postscript* format, the choice of postscript format as presentation format is easy. In contrast, the choice of a convenient presentation format for the *asynchronous* scenario is not as obvious. Two different objectives have to be pursued:

- Allocation of a print version of the transparencies used in the lecture. This allows students to annotate during the lecture.
- Allocation of a presentation format for the Web, to be integrated with the audio/video streams of the recorded lecture as well as supplementary material.

Both objectives differ in design requirements; thus we have to support more than one format. As postscript has to be prepared for the synchronous scenario, this format was initially chosen for printing. During the ongoing semester, students encountered problems in printing postscript and we decided to also make PDF format available.

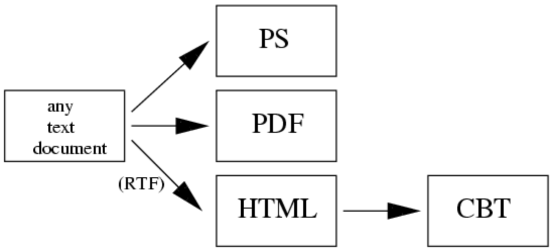


Figure 1: primary format, presentation format and CBT

But what is an appropriate format for the online CBT version? Not only shall the text be readable in this format but pictures and graphics, the audio and the video streams of the recorded lecture, and supplementary information shall be included. To support platform independence, the choice of a convenient presentation format is basically the choice between PDF and HTML. The advantage of PDF is certainly the existence of tools that perform a high quality conversion from different primary formats. A trait of HTML is its openness to other document formats. For example Java applets can be integrated seamlessly into HTML documents. A major advantage of HTML documents is that HTML can be post-processed using scripting languages e.g. to integrate additional hyperlinks. HTML is therefore the most appropriate presentation format for the CBT. Figure 1 visualizes the relationship of primary format and presentation formats in our system.

The Teachware Production Process

Our teachware production process allows all three different presentation formats (postscript, PDF, HTML) to be produced out of a single primary format. This process consists of three steps.

In the first step, the author creates her documents by help of her favorite authoring tool. In our case the lecturer uses Word. In the Summer semester 1999, a second department of the University of Mannheim joined our project; they use Latex as the primary format.

In the second step, the documents are converted into the three presentation formats. Standard conversion tools exist for the postscript and PDF format. Meanwhile the conversion from Latex to HTML is standardized, our experiments with HTML export filters of arbitrary word processing tools showed that the resulting HTML documents could hardly be post-processed. Furthermore the use of an internal export filter would limit the choice of word processing tools as they must provide such a filter. Very good results in HTML code were achieved by exporting documents into *Rich Text Format* (RTF) and converting the RTF document into HTML with the tool *rtf2html*. This process is suitable for all authoring tools that support an export function to RTF and therefore does not restrict the flexibility of tool choice. Finally, two print formats (postscript and PDF) and an HTML document result from this second processing step.

The third step only concerns our HTML document and generates the integrated CBT course. The HTML document resulting from step two contains a textual presentation of the primary documents. This is inappropriate for presentation and neither meets our goal of a multimedia document that includes video, audio, full text, animation and exercises nor does it support the key of Web-technology, linking. Here a Perl script does most of the work. We will have a closer look on it later on.

GENERATING HYPERLINKS AUTOMATICALLY

Instructional Design

In cooperation with a research group at the Department of Psychology at the University of Heidelberg an *Instructional Design* for the CBT was developed. The slides used during a live lecture form the basis of the CBT. They are linked to each other in the order of their presentation by the lecturer. Supplementary to this *first level information* - clearly ordered one after the other – *further levels of information* can be added in vertical order in a structuring tree (Figure 2).

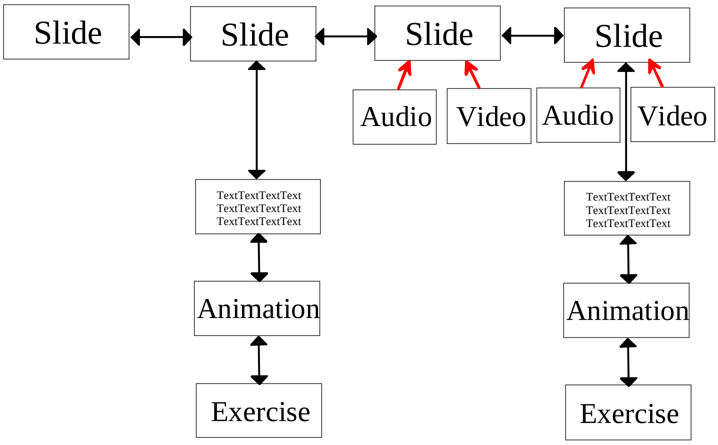
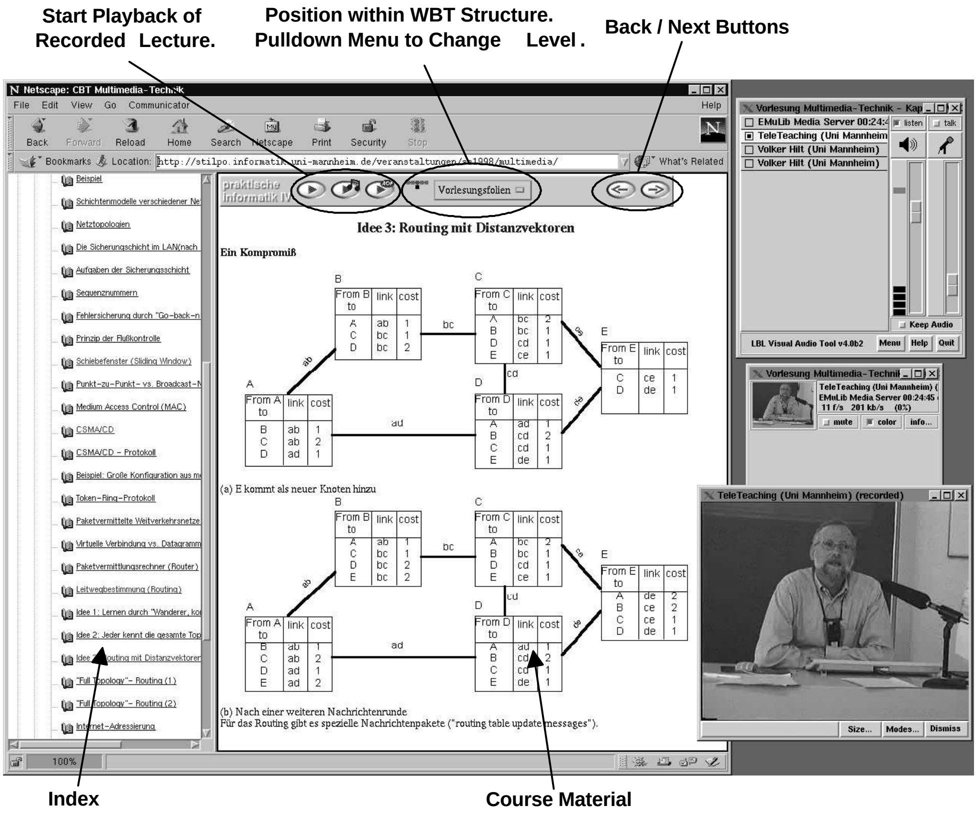


Figure 2: Instructional Design for our CBT Course

To build the multimedia document, video and audio data are added to each slide. The topics discussed on a slide can be elaborated with additional text, animations or exercises. For example, processes with a dynamic behavior over time are hard to explain with static representation such as graphics in textbooks. This sometimes results in cartoon-like images, each one representing a precise point in time. As a more appropriate didactical medium we have developed a number of animations and simulations. They are written in Java to ensure compatibility to all platforms. As shown in Figure 2 these Java applets are also added as a lower layer to a slide. To visualize the linking of the slides as well as of the supplementary information, a *navigation bar* is provided on the top of each HTML page. The navigation bar contains buttons for forward and backward linking. Other buttons start the playback of recorded video and audio streams. A little icon shows whether supplementary information exists for this slide, and a pulldown menu allows it to be displayed (Figure 3). For easy orientation and random access within the CBT, a self-extracting table of contents in an extra frame completes the design of our screen.

Linking Technology

The Instructional Design presented above requires the generation of horizontal links between the slides as well as vertical links to supplementary material. Not only is the number of links very high, but the structure of the links is very complex so that it is desirable not to maintain them by hand.


Start Playback of Recorded Lecture.
Position within WBT Structure. Pulldown Menu to Change Level.
Back / Next Buttons
Netscape: CBT Multimedia-Technik
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks Location: http://stilpo.informatik.uni-mannheim.de/veranstaltungen/ss1998/multimedia/
What's Related
praktische informatik IV
Vorlesungsfolien
Vorlesung Multimedia-Technik
EMuLib Media Server 00:24:4
TeleTeaching (Uni Mannheim)
Volker Hilt (Uni Mannheim)
Volker Hilt (Uni Mannheim)
listen talk
Keep Audio
LBL Visual Audio Tool v4.0b2
Menu Help Quit
TeleTeaching (Uni Mannheim)
EMuLib Media Server 00:24:45
11 f/s 201 kb/s (0%)
mute color info...
TeleTeaching (Uni Mannheim) (recorded)
Size... Modes... Dismiss
100%
Index
Course Material

Beispiel
Schichtenmodelle verschiedener Ne
Netztopologien
Die Sicherungschicht im LAN(nach
Aufgaben der Sicherungsschicht
Sequenznummern
Fehlersicherung durch "Go-back-n
Prinzip der Flußkontrolle
Schiebefenster (Sliding Window)
Punkt-zu-Punkt- vs. Broadcast-N
Medium Access Control (MAC)
CSMA/CD
CSMA/CD - Protokoll
Beispiel: Große Konfiguration aus m
Token-Ring-Protokoll
Paketvermittelte Weitverkehrsnetze
Virtuelle Verbindung vs. Datagramm
Paketvermittlungsrechner (Router)
Leitwegbestimmung (Routing)
Idee 1: Lernen durch "Wanderer, ko
Idee 2: Jeder kennt die gesamte Top
Idee 3: Routing mit Distanzvektoren
"Full Topology"- Routing (1)
"Full Topology"- Routing (2)
Internet-Adressierung

Idee 3: Routing mit Distanzvektoren

Ein Kompromiß

A

From A to	link	cost
B	ab	1
C	ab	2
D	ad	1

B

From B to	link	cost
A	ab	1
C	bc	1
D	bc	2

C

From C to	link	cost
A	bc	2
B	bc	1
D	cd	1
E	ce	1

D

From D to	link	cost
A	ad	1
B	cd	2
C	cd	1
E	de	1

E

From E to	link	cost
C	ce	1
D	de	1

ab bc cd ad ce de

(a) E kommt als neuer Knoten hinzu

A

From A to	link	cost
B	ab	1
C	ab	2
D	ad	1
E	ad	2

B

From B to	link	cost
A	ab	1
C	bc	1
D	bc	2
E	bc	2

C

From C to	link	cost
A	bc	2
B	bc	1
D	cd	1
E	ce	1

D

From D to	link	cost
A	ad	1
B	cd	2
C	cd	1
E	de	1

E

From E to	link	cost
A	de	2
B	ce	2
C	ce	1
D	de	1

ab bc cd ad ce de

(b) Nach einer weiteren Nachrichtenrunde
Für das Routing gibt es spezielle Nachrichtenpakete ("routing table update messages").

Figure 3: Screen-shot of the CBT

The Perl script which is run at step three in the teachware production process automatically sets up the CBT structure from our basic HTML pages. To accomplish this task, the script reads a configuration file in which the linking structure is stored. The content of this configuration file is automatically generated during the live lecture and can be modified by the author. It comprises the titles of the HTML pages, indexes for video and audio data and information about corresponding supplementary information.

How do we know which parts of the audio and video streams belong to a single HTML page? We already mentioned that the recording of audio and video is performed by the MBone VCR on the Multicast-Backbone of the Internet. The VCR is based on the RTP protocol, the same protocol used by the video conferencing tool *vic* and the visual audio tool *vat* for data transmission. In contrast to vic and vat, the MBone whiteboard *wb* works on a proprietary protocol to accomplish reliable multicast transmission. Therefore we had to face the recording problem of VCR being unable to record wb data streams. Since kowledge about „when a page in the wb is turned" during the speech would overcome this problem, we have implemented the tool *listenwb* for video and audio indexing in order to determine the parts of video and audio belonging to a single slide. The tool records the time at which a page is turned in the wb. These times are converted into timestamps relative to the start time of the audio and video recording. At the end of a recording, an index of all timestamps is created.

Figure 4 presents a configuration file read by the Perl script to automatically create the links within our CBT. The titles of the pages are given in the beginning of a line (e.g. „5.4 Wegewahl (Routing) für Multicast-Netze“). Two numbers follow the titles (e.g. „1 267“). They are the timestamps for the beginning and ending of audio and video streams to the according page relative to the recording of that lecture. If there is supplementary information for a page, the URL and the description are added (e.g. „anim_dvmrp_index.html“ „DVMRP Animation“). Adding URL and title of supplementary information is the only thing still performed with manual control in our teachware production process.

```
                 :                                   :     :
„5.4 Wegewahl (Routing) f&uuml;r Multicast-Netze“    1     267
„Motivation f&uuml;r Multicast“                      267   766
                 :                                   :     :
„Routing-Algorithmen f&uuml;r Multicast“             2565  2970
      „uebungen/multicastrout.html“ „&Uuml;bungsaufgabe“
„Reverse Path Broadcasting“                          2970  3165
      „uebungen/rpb.html“           „&Uuml;bungsaufgabe“
                 :                                   :     :
„Multicast-Routing im Internet“                      4693  4800
      „anim_dvmrp_index.html“       „DVMRP Animation“
      „uebungen/dvmrp.html“         „&Uuml;bungsaufgabe“
```

Figure 4: Configuration File read by the Perl Script

EXPERIENCE

Digital video and audio streams included in our computer-based training material had to be recorded during the live lecture. As we have transmitted the lectures *Multimedia Technology* and *Computer Networks* in the synchronous teleteaching scenarios to remote lecture rooms, a camera and a microphone were already installed in the lecture rooms and minimal additional effort was required for recording. The transmission of lectures however is in no case necessary for our process.

After the development of the tools Mbone VCR, listenwb and the Perl script had been accomplished, a second recording problem had to be faced. On the average, 600 kbit/s are used for video transmission during a live lecture. The recording of a 90-minute lecture would devour 395 MB. In computer science lectures, the video usually shows the talking head of the lecturer and serves mostly as a psychological tool for personalization. High video quality is therefore not essential. After we scaled down the video rate to 128 kbit/s each 90-minute lecture consumed a disc space of 84 MB which is acceptable. The scaling of the video data is done using the *rtp-gateway* [AMZ95].

One year of experience in recording and processing live lectures has also exposed some disadvantages. As speech is live, there is always some „äh“ and „hm“ to be found, or remarks on the weather or the number of students in attendance. These remarks don’t make sense when listening to them weeks later at a remote location. But students told us that it does not disturb them and that it lives up the atmosphere. There are deliberations about post-processing the audio-clips. But this would entail a complete restructuring of the MBone VCR. We doubt any measurable impact on our goal, which is to allow the students a remote and time-independent learning with full quality assurance - all with minimal effort.

Our Department of Applied Computer Science at the University of Mannheim has experience with synchronous teleteaching since 1995. The lectures from this department were the first ones to practice the recording. The resulting CBTs *Multimedia Technology* can be found at

[Eff98, Eff99]. In the Winter semester 1998/99, the CBT *Computer Networks* was produced and is stored at [Eff9899].

CONCLUSIONS AND OUTLOOK

Starting with the only postulation that a digital form of the material used for a lecture should already exist (which proves to be correct in almost all cases), we pointed out how to generate a computer-based training out of this material. The CBT proposed can gradually be enlarged with audio, video, animations, exercises and additional text, just as the author likes. The creation process is automated by a Perl script; only the configuration has to be edited manually.

During the Summer semester 1999, the Department of Information Systems at the University of Mannheim has joined the described process for creation of a CBT version for the lecture on Java technology. Here, the lectures are prepared using the primary format Latex [Scha99].

In the ongoing Summer semester, the lecture Multimedia Technology is held and recorded the second time. When the lecturer has had to be absent due to a conflicting commitment elsewhere, common practice to date has been to ask someone else to substitute. In this semester we are planning to experimentally play back the „canned goods“ and evaluate how our students react.

Though we have asked single students about their opinion and experience with the three CBTs created until today, we have not yet systematically evaluated the acceptance of our product. The Departmentof Psychology at the University of Mannheim has offered help to realize a thorough questionnaire in the upcoming semesters.

Acknowledgements. We would like to thank Jörg Wittmer who has developed the Perl script that forms the core of the described processes. The projects at the University of Mannheim dealing with teleteaching and asynchronous learning are partly financed by the DFG (Deutsche Forschungsgemeinschaft) in the project „V3D2 Digital Library Initiative“ and the Land Baden-Württemberg in the project „VIROR. Virtuelle Hochschule Oberrhein“.

REFERENCES

[AMZ95] Elan Amir, Steven McCanne, and Hui Zhang. “An Application-level Video Gateway.” In *Proc. ACM Multimedia ‘95*, San Francisco, pages 255 - 265, November 1995.

[Ber99] FU Berlin. “Dialekt. Digitale interaktive Lektionen in der Studentenausbildung.” *http://dialekt.cedis.fu-berlin.de/*, 1999.

[BMO97] Christian Bacher, Rainer Müller, and Thomas Ottmann. “Ein Weg zur Integration von Live-Vorlesung, Teleteaching und Lehrsoftwareproduktion.” *Mediengestützte wissenschaftliche Weiterbildung.* Braunschweig, 1997. [in German].

[BMOW97] Christian Bacher, Rainer Müller, Thomas Ottmann, and M. Will. “Authoring on the Fly.” A new way of integrating telepresentation and courseware production. *ICCE*, 1997.

[Dar99] TU Darmstadt. Multibook. “A Multimedia Learn- and Teach-Book.” *http://www.multibook.de*, 1999.

[Dre98] TU Dresden. JaTeK. “Java Based Teleteaching Kit.” *http://telet.inf.tu-dresden.de/*, 1998.

[Eff97] W. Effelsberg. “Das Projekt TeleTeaching der Universitäten Mannheim und Heidelberg.” In *Proc. LEARNTEC ‘97*, Karlsruhe, Germany, January 1997. [in German].

[Eff98] Wolfgang Effelsberg. “Multimedia Technology. Course given at the University of Mannheim.” *http://www-mm.informatik.uni-mannheim.de/veranstaltungen/-ss1998/multimedia/*, July 1998. [in German].

[Eff9899] Wolfgang Effelsberg. “Computer Networks. Course given at the University of Mannheim.” *http://www-mm.informatik.uni-mannheim.de/veranstaltungen/-ws199899/rechnernetze/*, February 1999. [in German].

[Eff99] Wolfgang Effelsberg. “Multimedia Technology. Course given at the University of Mannheim.” *http://www-mm.informatik.uni-mannheim.de/veranstaltungen/-ss1999/multimedia/*, July 1999. [in German].

[EGE97] Andreas Eckert, Werner Geyer, and Wolfgang Effelsberg. “A Distance Learning System for Higher Education in Telecommunications and Multimedia - A Compound Organizational, Pedagogical and Technical Approach.” *In Proc. ED-MEDIA/ED-TELECOM ‘97*, Calgary, Canada. Association for the Advancement of Computing in Education (AACE), 1997. [on CD-ROM only].

[Hag99] Gesamthochschule Hagen. “Die Virtuelle FernUniversität.” *http://vu.fernuni.hagen.de/feuvus.f.html*, 1999.

[HK99] Volker Hilt and Christoph Kuhmünch. “New Tools for Synchronous and Asynchronous Teaching and Learning in the Internet.” *In Proc. ED-MEDIA/ED-TELECOM ‘99*, Seattle. Association for the Advancement of Computing in Education (AACE), 1999. [to appear].

[Hol97] Wieland Holfelder. “Interactive Remote Recording and Playback of Multicast Videoconferences.” In *Proc. 4th International Workshop on Interactive Distributed Multimedia Systems and Telecomminication Services (IDMS) ‘97*, volume 1309, pages 450-463. Lecture Notes in Computer Science (LNCS), Springer-Verlag, Heidelberg, September 1997.

[Scha99] Martin Schader. “Java Technology. Course given at the University of Mannheim.” *http://www-mm.informatik.uni-mannheim.de/veranstaltungen/test/-java/*, July 1999. [in German].