

A METHOD FOR ASYNCHRONOUS, WEB-BASED LECTURE DELIVERY

Roberto Almeida Bittencourt^{*1} and David A. Carr²

Abstract $\frac{3}{4}$ Computer-based distance education has been growing enormously in recent years due to advances in key technologies: telecommunications, computer technology, graphical user interfaces, and the Internet. Distance education materials can be presented either synchronously, creating the illusion of a classroom on the computer, or asynchronously, allowing the students access-on-demand. We describe the development of a system to solve part of the asynchronous education problem – delivery of pre-recorded multimedia lectures via a web browser. We built a demonstration system using off-the-shelf technology making it possible to make presentations using commercial tools and to display them on standard systems. We designed the system with lectures composed of “slides” accompanied by audio commentary. These presentations can be delivered via low-speed modem links – an important consideration in places without a well-developed infrastructure. We also believe that computer-based education is often so complicated to use that operation interferes with education. In order to avoid this, we evaluated a prototype system to obtain feedback about difficulties in operation and to correct them.

Index Terms $\frac{3}{4}$ Distance education, multimedia, usability evaluation

INTRODUCTION

Computer-based distance education has been growing enormously in recent years, primarily due to advances in key technologies: telecommunications, computer technology, graphical user interfaces, and the Internet. Distance education materials can be presented either synchronously or asynchronously. The synchronous method strives to create the illusion of a classroom on the computer. The asynchronous method allows the students to access stored educational material on demand. In asynchronous systems, students follow their own pace and may access the course materials independent of other students and at different times.

This paper describes the development of a system to solve one aspect of the asynchronous education problem, delivering previously recorded multimedia lectures via a web browser. We built a demonstration system, which integrates Java, JavaScript, SMIL, HTML, and Real Networks streaming media. This combination makes it possible to build multimedia presentations using commercial tools and display them on standard systems. Furthermore, we

designed the system with the intent that a lecture would be composed of “slides” and an audio commentary. These presentations can be delivered via low-speed modem links, which can be important in areas of the world without a well-developed infrastructure.

We believe that computer-based educational systems are often so complicated to use that operation interferes with education. In order to avoid this, we employed a technique called “Cooperative Evaluation” [12]. A cooperative evaluation is performed by having a potential user of the system perform a number of tasks with the system. At the same time, one or more system designers observe, paying special attention to where the user has difficulty. In order to enrich the information, users are asked to describe what they are thinking. We created a prototype course explaining the basics of SMIL and used it to perform a cooperative evaluation with each of five students.

RELATED WEB-BASED LEARNING SYSTEMS

Different systems have been developed to provide web-based instruction. Some, like ClassPoint [18], are fully synchronous. ClassPoint was developed by White Pine Software from CU-SeeMe. It integrates videoconferencing with web navigation. It provides an instructor-controlled learning environment with multi-point video and audio, instructor-led Web Tours, Web-based class setup and scheduling, shared whiteboards, text chat, and other tools. Although the instructor leads the conference, all conference members can participate equally. This mode requires high network bandwidth to provide multi-point video and audio. ClassPoint can be configured for less bandwidth by having only the teacher broadcast audio and video, while the students can use just the text chat. This however reduces the already limited sense of presence in the classroom. Network lags can also disturb the broadcast, creating a feeling of lack of continuity.

Stanford Online, from Stanford University, is aimed at both synchronous and asynchronous learning. It provides video and audio streaming synchronized with a slideshow. It also provides some additional tools such as a bulletin board and a chat room [3, 7, 16]. As the student watches the video of the lecture in a smaller window, the slides change automatically. During recorded lectures, the student can pause to post a question to the course bulletin board. There are also indices in the video window for skipping to specific

^{*} Financed by a grant in aid from the Swedish Foundation for International Cooperation in Research and Higher Education — STINT.

¹ Roberto Almeida Bittencourt, Universidade Estadual de Feira de Santana – UEFS, Dpto. Ciências Exatas, Mód. V, Km 03 - BR 116 - Campus Universitário, 44.031-460 Feira de Santana, BA, Brazil, robert@uefs.br

² David A. Carr, Institutionen för Systemteknik, Luleå Tekniska Universitet, S-971 87 Luleå, Sweden, david@sm.luth.se

points of the lecture; however, these indices are tied to the temporal structure and not to content. So, searching based on content is not available. In order to adjust the presentation to a range of connections, low and high bandwidth versions of the courses can be provided.

mMOD [10, 14] is another system with both synchronous and asynchronous modes. In the asynchronous mode, it provides a lecture player where a recorded lecture can be replayed. mMOD uses multicasting to deliver the course, and thereby, reduces the network load for sessions where many participants are on the same subnets. From the author's point-of-view, the mMOD system requires minimal extra effort for instructors. They produce lecture notes and convert them to HTML before the lecture. They then deliver the lecture, and it is recorded for later playback. However, students have only a VCR-metaphor control and cannot randomly access arbitrary parts of the lecture.

WebCT [5, 17] from the University of British Columbia, is currently one of the leading tools for asynchronous web-based courses. It provides an integrated environment for course notes, announcements, and assessment. WebCT has a simple web interface – both for the course creator and for the student. It organizes the contents hierarchically. Students can then go directly to a specific point using the hierarchy, a search tool, a glossary or an index. WebCT is a more productive way to create course web pages. However, it is primarily HTML-based and does not provide explicit support for multimedia. WebCT simplifies tracking student progress through mechanisms for monitoring and evaluation.

AulaNet [2, 8, 9] from the Catholic University of Rio de Janeiro is aimed mainly at asynchronous education and is strongly based on the idea of collaboration. It can either use teleconferencing with a CU-SeeMe tool or streaming video and audio for recorded lectures. This is a value added to the conventional course notes. The drawback in AulaNet is that there is no synchronization mechanism between the video recording and other course materials such as slides or other course notes. AulaNet also lacks localization mechanisms such as tables of contents, search tools, or indices.

AN ASYNCHRONOUS LECTURE DELIVERY SYSTEM

All of the above systems either did not explicitly support multimedia, did not allow flexible access to content, or contained some proprietary elements. We decided to see if we could design a system that supported distance education without these drawbacks. We wanted to include lectures as a component of the course because they are an efficient way of delivering information about a subject. We also believe that it should be possible to create part of a distance course by recording during classroom-based courses and that enhancement of these courses is a way to reduce the cost of creating distance courses.

We quickly realized that we could not produce an entire distance education system at once. But, we felt that an

asynchronous lecture delivery system was a good first step. In addition to asynchronous delivery, the system had to meet four primary goals:

1. **Affordability.** The environment should be cost free to end-users. The cost of authoring tools should also be minimal.
2. **Low extra effort.** It should take little or no effort to install, preferably using components already on the user's system. Ideally, authoring should be no more difficult than preparing a regular classroom lecture.
3. **Low bandwidth.** The system should operate on dial-up lines with 28.8K modems as it was anticipated that the system would be used in locations where dedicated, high-speed Internet connections would not be available. It should also be able to use higher bandwidth if available.
4. **Standard tools and protocols.** The system should allow the use of existing software and protocols for the Internet.

These goals impose constraints on what our system can deliver and what the base platform can be. First, the low-bandwidth requirement rules out large blocks of video. This means that a video image of the instructor cannot be delivered. We did not feel a video of the instructor was a strong requirement of the system. Our experience indicated that audio and audio-visual aids such as overhead transparencies were much more important. In addition, we conducted a survey of 71 students and asked them to rank the most important potential features for a system. Video of the teacher was ranked nineteenth of twenty-three features. The five most important features were:

1. The ability to stop the lecture and skip back-and-forth through the presentation.
2. Synchronization between presentation slides and the audio (and/or video) of the teacher.
3. A hierarchical table of contents that allowed quick access to any portion of the presentation.
4. Presentation slides explaining the lecture content.
5. Audio of the teacher.

Other features that the students thought important can be summarized as flexibility of access (search over an index, presentation of one slide at a time) and enhanced content (simulations and additional explanations linked to the slide).

The affordability, low extra effort, and support for standards goals mean that the base platform should be a web browser enhanced if necessary by downloadable, free plug-ins. As we wanted to deliver audio synchronized with slides (web pages), we needed to choose a widely available, web-based multimedia format. Streaming multimedia is an ideal way to deliver pre-recorded lectures. At the time we were developing our system, there were the three competing systems available: Real Networks [15], Microsoft Windows Media [11] and Apple Quick Time [1]. In order to make a choice, we evaluated the systems in three areas: the server,

the client, and authoring tools. For servers, we found that the Real Networks solution offered the most flexibility except for platform support. QuickTime was supported on more platforms, but a number of the servers were beta versions. We evaluated the client in five areas: supported platforms, supported browsers, supported protocols, ease of control by the browser, and cost. Ease of control by the browser was especially important for developing the user interface. In this category, the real difference between systems was that Windows Media lacked a well-documented control interface for either Java or JavaScript. Authoring tools was the final area of comparison. From our evaluation, we found that Windows Media and Real Networks had good, low-cost authoring tools. Based on this comparison we selected Real Networks. We might have chosen Windows Media on the basis of cost, if it had had a well-defined external control interface and a server that could be run on a platform other than the Server Edition of Windows NT.

System Design

Streaming multimedia alone is not enough to deliver an asynchronous lecture. Students need to find courses and lectures. They need to find the section of the lecture that interests them, and they need to control the presentation. In

addition, each system supplier provides only part of the solution. Thus, we developed a system in which a database held both course and student information, a web server bound together various components supporting navigation among them, and a streaming multimedia server provided the lectures. On the client side the web browser was central. But, an interface to the database and control of the multimedia player were needed. Figure 1 shows an overview of the system's architecture.

Our choice of off-the-shelf components led to a rather complex solution involving four different languages: HTML, the Synchronized Multimedia Integration Language (SMIL) [19], JavaScript [4, 13], and Java. HTML provided the ability to embed audio, video, and animation, to bind together various components, and to navigate among them. SMIL has markups (or tags) for controlling the content to be presented (preferred language, bandwidth available on the player), specifying the layout of images and video on the screen, and describing the types of media elements (audio, video, images, text, text streams and animation). It also has tags for hyper-linking and for handling timing and synchronization. Together with HTML, SMIL provided a framework that allowed flexible and sophisticated presentations on the Web. JavaScript provided an interface to the Real Player plug-in via a control API. JavaScript also

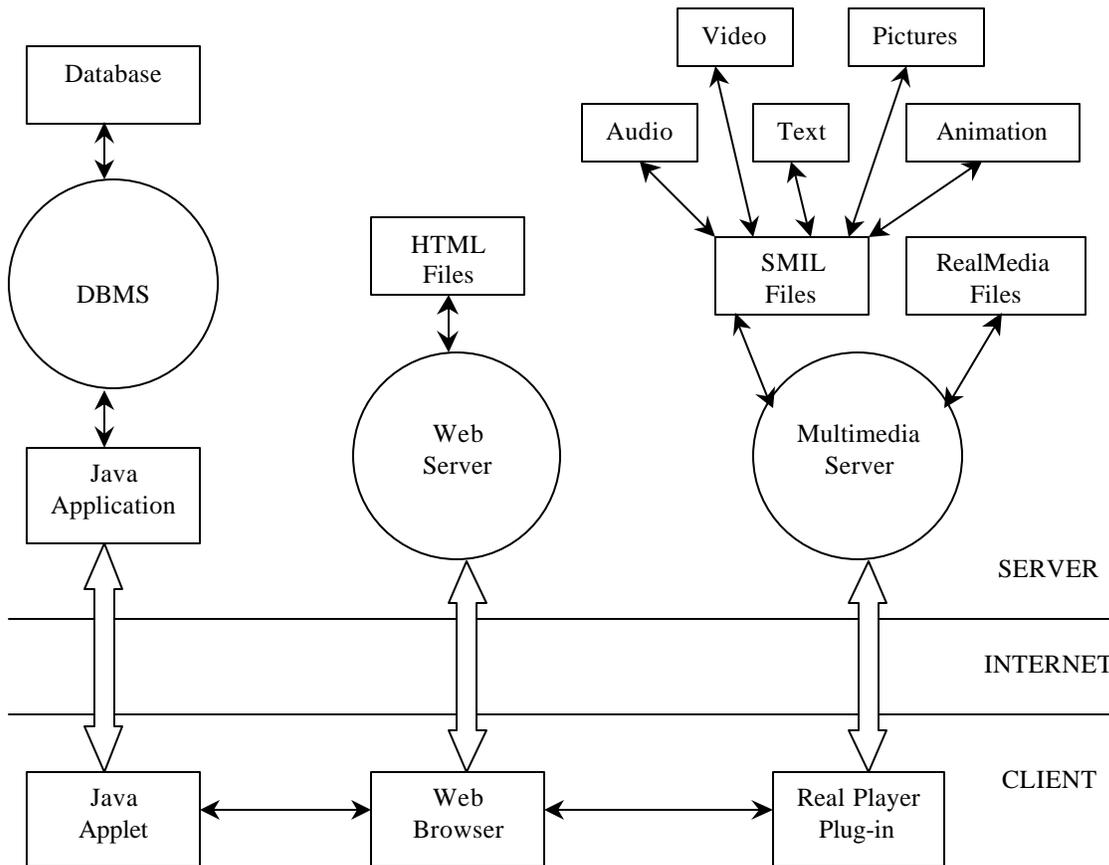


FIGURE 1
ARCHITECTURE OF THE LECTURE ENVIRONMENT

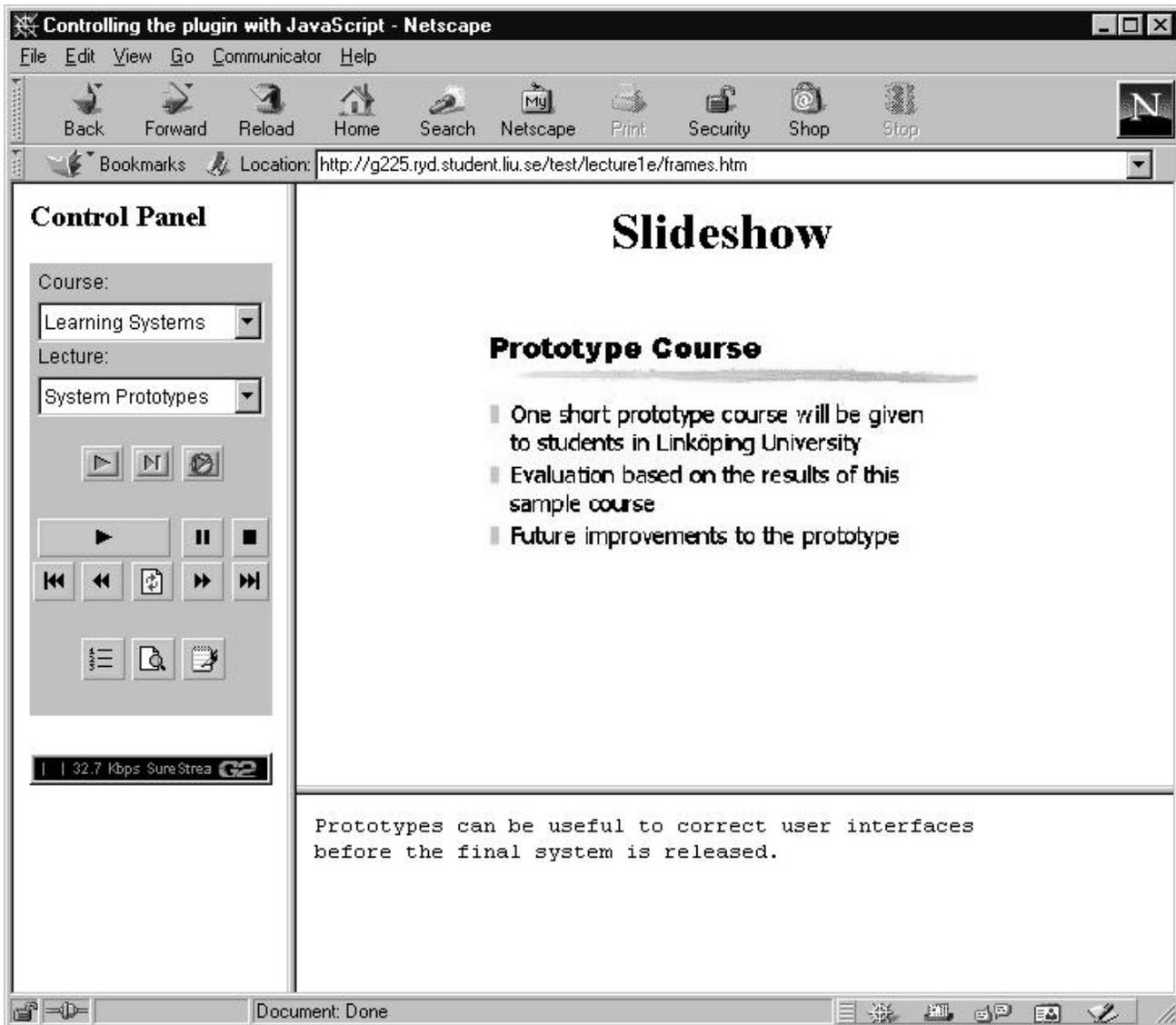


FIGURE 2
FINAL LAYOUT OF THE LECTURE SYSTEM

has an interface with the Java language called LiveConnect. Through LiveConnect, it is possible for an applet to execute JavaScript commands and for a script in JavaScript to execute methods from Java objects. We exploit this in order to control the streaming media (and thus, the lecture sequencing). Finally, our system uses a Java applet as the glue between various components. The applet opens a TCP/IP connection to a database serving application in order to access the course and student database. The applet also provides the user interface and, through JavaScript, controls the streaming media's playback.

Overview of the System

In order to hand over the control of the learning process to the student, three different presentation modes were devised:

- a continuous flow, which works like a VCR;
- an interrupted flow, where the slideshow pauses after the presentation of each slide;
- a preview mode, where only the slides of the course are shown without audio.

The system itself is presented on a web page that is divided into three frames. (See Figure 2.) The controls for the presentation are in the left frame, the presentation slides are in the upper-right frame, and an area for student notes is in the lower-right frame.

After choosing the desired course and lecture from the list boxes, the student can start the lecture in any of the three presentation modes previously described, using the buttons located just below the list boxes. According to the chosen mode, the remaining buttons will work in a slightly different

fashion, but keeping the same general function expressed in their icons.

In the continuous-flow mode, the “Play”, “Pause”, “Stop”, “First”, “Previous”, “Replay”, “Next”, and “Last” buttons work in a way similar to a CD player, where each slide and its associated audio commentary can be seen as a CD audio track.

In the interrupted-flow mode, the behavior is similar, except that after each audio commentary the presentation pauses, giving the student time to take notes.

In the preview mode, “Play”, “Pause”, and “Stop” buttons are disabled, no audio is heard, and only the slides are shown. The other buttons can be used to skip back and forth through the slides.

The three remaining buttons at the bottom open a table of contents, a search tool, and a notepad. All of these come in a small browser window that does not hide the slides. The table of contents provides links for hierarchical organization. When clicked, they work very much in the same way as the “Previous” and “Next” buttons, except that the student can now skip to any point in the lecture or even to another lecture by choosing the desired link. The search tool allows searching for a keyword and returns links to slides that contain the keyword. These links work in the same way as in the table of contents. Finally, the notepad can record a written comment in the course database associated both to a slide and to the student. This comment is then shown together with the slide in the lower-right frame of the main window.

Breaking the linear access approach was one aim in the design. This more flexible access provided a partition of the lecture into knowledge units, so that the student could choose what to learn and also to skip content considered unnecessary in the same way one does with a book.

Cooperative Evaluation of the System

In order to be effective, a lecture delivery system must support students and not distract them from the educational material. The best way to insure this is to test the design with representatives of the user community — in this case, students. We used a method called “Cooperative Evaluation” [12]. For a Cooperative Evaluation one recruits from 35 users, called evaluators, to assist in the evaluation by operating the system. Cooperative Evaluation works as follows:

- The designers construct a set of representative tasks for the evaluators to perform. These tasks should be described in sufficient detail so that the evaluators understand what is needed, but the manner in which the tasks are performed should not be described. An important part of the test of the system is to see if the evaluators can discover how to use it to accomplish the tasks.
- The evaluators receive a brief explanation of how the system works, and then, start using the system.

- During system use, the evaluators are encouraged to “think aloud”. Because it is difficult to know what evaluators are thinking when they operate the system and this information is useful for refining the design, evaluators are asked to describe what they are doing, what they are planning to do next, what they are looking for, etc. This is called thinking aloud. The designer should encourage the evaluator to think aloud without disrupting things and pay special attention when evaluators fall silent, as this can be an indication of confusion or trouble in operating the system. Evaluators are also encouraged to ask questions when doubtful. A Cooperative Evaluation should not be seen as a test of the system, per se, but rather it should be seen as a dialog between designers and evaluators that is used to discover problems and refine the system design.
- After evaluators have finished using the system, they should be debriefed. The designer should go over problems that were encountered. Evaluators should be asked to discuss their general impressions of the system. They can also make suggestions for improvements, but are not required to do so.

In order to evaluate the effectiveness of our system, we created a prototype course explaining the basics of the SMIL language. Using this course, we performed a cooperative evaluation with five students recruited at Linköping University. All of the students were experienced with computers, including programming. They had a background in either Computer Science or Electrical Engineering. The test group used the system in computers either at home or at the university. One of them used a 56 KBPS modem to access the lecture system. The other four had direct access to the university’s local network. The object of the evaluation was to find areas where the evaluators had trouble with the interface rather than to demonstrate any properties of the system.

In addition to exposing shortcomings of the user interface, the evaluation also demonstrated positive aspects. The evaluators liked such features as: the ability to choose what to learn, the ability to choose how the material is presented (through three different modes), and the possibility to repeat something that was unclear. This makes the system more flexible and interactive in comparison with other previously developed asynchronous multimedia systems. However, no feedback is provided (e.g.: a bulletin board, intelligent tutoring), and it is not possible to choose in advance the content to be seen, i.e., customized courses. An integration of this system to other tools is essential to make it reasonably complete for distance-education use.

Experience with Authoring

Our experience with creating the course revealed that the dynamics of communication offered by the system are very good. The teacher can choose from a variety of media and combine them in a single presentation. However, there is

still the authoring problem to be solved. Creating a presentation with synchronized slides and audio is simple and better than with some other systems. But, adding videos, text streams, and animation requires using more sophisticated tools that seem too complex for most course authors.

Course preparation is estimated to take 5 to 10 times the time required for the preparation of a set of overheads for the same presentation length. The extra work comes largely from three sources: extra care taken when recording, determining the timing of slides for indexing, and inserting indexing information in the course database.

Further decreases in the preparation time can be achieved with the development of tools for marking the time slots, automatically updating the database, and a better tool for indexing the text of the slides.

CONCLUSIONS

This work showed that it is possible to combine flexibility of access to content and a rich variety of media in asynchronous web-based learning systems. It showed also that a broad range of users can be reached using standard software components present on personal computers like web browsers, Java applets, and multimedia plug-ins. Nevertheless, it was also apparent that the integration of these standard software components is still in an early phase and that imperfections arise when they are used in combination. Consolidation of these technologies will make it possible to use the kind of system resulting from this work on almost any kind of computer connected to the Internet.

This system provides a more flexible environment for delivering lectures for both distance and classroom learning. However, the system developed here is just a prototype and, as such, can be improved. Besides improvements to the present system, some research and development can be performed in at least three different areas:

1. Development of tools to make course creation easier. We particularly think that recording a lecture as it is delivered in the classroom and automatically indexing it would provide a low-effort way to develop the basics of an electronic course.
2. Test of the system in real-world situations and evaluation of the design.
3. Adding new tools to the web-based lecture system — particularly assessment tools and a way to ask questions.

ACKNOWLEDGMENT

We would like to thank those who participated in the Cooperative Evaluation even though they must remain anonymous. We would also like to thank Carl Rollo for proofreading our drafts. The work described in this paper was performed while both authors were at Linköping University in Linköping, Sweden.

REFERENCES

- [1] Apple Computer, Inc., *Quick Time*. <http://www.apple.com/quicktime/>, accessed 15-Oct-1999.
- [2] Catholic University of Rio de Janeiro, AulaNet. <http://ead.les.inf.puc-rio.br/aulanet/>, accessed 27-Sep-1999.
- [3] Cordero, C., Harris, D., & Hsieh, J., High speed network for delivery of education-on-demand. Proceedings of SPIE 2667 – Multimedia Computing and Networking, 1996.
- [4] Flanagan, D., *JavaScript: The Definitive Guide*. 3rd ed., O'Reilly, Sebastopol, CA, 1998.
- [5] Goldberg, M. W., & Salari S., An Update on WebCT (World-Wide-Web Course Tools) - a Tool for the Creation of Sophisticated Web-Based Learning Environments. Proceedings of NAUWeb '97 – Current Practices in Web-Based Course Development, Flagstaff, Arizona, June 1997.
- [6] Gundavaram, S., *CGI Programming on the World Wide Web*. O'Reilly, Sebastopol, CA, 1996.
- [7] Harris, D., DiPaolo, A., Advancing Asynchronous Distance Education Using High-Speed Networks, *IEEE Transactions on Education*, Aug 1996.
- [8] Laufer, C., Fuks, H., & Lucena, C. J. P., Rio InternetTV - AulaNet: Using Videoconference in Web-based Learning. Proceedings of WebNet'98 - World Conference of the WWW, Internet & Intranet. Association for the Advancement of Computing in Education, Charlottesville, VA, 1998.
- [9] Lucena, C. J. P., Fuks, H., Milidui, R., Laufer, C., Blois, M., Choren, R., Torres, V., & Daflon, L., AulaNet: Helping Teachers To Do Their Homework. Proceedings of Multimedia Computer Techniques in Engineering Education, Graz, Austria, Feb 1999.
- [10] Luleå University of Technology, mMOD – Recording and Media-on-Demand. <http://mates.cdt.luth.se/software/mMOD/>, accessed 27-Sep-1999
- [11] Microsoft Corporation, *Windows Media Home*. <http://www.microsoft.com/Windows/windowsmedia/>, accessed 14-Oct-1999.
- [12] Monk, A., Wright, P., Haber, J., Davenport, L., *Improving your Human-Computer Interface: A Practical Technique*. Hemel Hempstead: Prentice Hall, 1993.
- [13] Netscape Communications Corporation, *Client-Side JavaScript Guide*. <http://developer.netscape.com/docs/manuals/js/client/jsguide/index.htm>, accessed 20-Oct-1999.
- [14] Parnes, P., Mattson, M., Synnes, K., & Schefström, D., mMOD: the multicast Media-on-Demand system. <http://www.cdt.luth.se/~peppar/progs/mMOD/WebNet97/mMOD.pdf>
- [15] RealNetworks, Inc., *RealNetworks Documentation Library*. <http://service.real.com/help/library/index.html>, accessed 11-Oct-1999.
- [16] Stanford University, Stanford Online, <http://stanford-online.stanford.edu/>, accessed 27-Sep-1999.
- [17] University of British Columbia, WebCT – World Wide Web Course Tools. <http://homebrew.cs.ubc.ca/webct/>, accessed 27-Sep-1999.
- [18] White Pine Software, *ClassPoint Product Information*. <http://www.wpine.com/Products/ClassPoint/>, accessed 27-Sep-1999.
- [19] W3C - The World Wide Web Consortium, *Synchronized Multimedia*. <http://www.w3.org/AudioVideo/>, accessed