Distributed Collaborative Learning with Open Source Software

Ulrich BORGOLTE and Andreas BISCHOFF

Department of Electrical Engineering and Information Technology,
University of Hagen
Universitaetsstr. 27, 58084 Hagen, Germany
Email: Ulrich.Borgolte@fernuni-hagen.de

Abstract:
This paper presents an approach for synchronous, online, and interactive collaboration in open and distance learning (ODL) environments. The approach significantly reduces the costs for students in remote locations, allowing them to participate in synchronous events even if resources are limited. With the methodology proposed, quality of ODL can be enhanced. Students in less-favoured regions will be enabled to participate in high-quality ODL.

Keywords:
Distance teaching, collaborative learning, synchronous events, open source

1. INTRODUCTION

Traditionally, higher education is based on synchronous communication, on real-time discussions of a certain topic. On the other hand, conventional ODL is based on asynchronous (e.g. on written courses, email, newsgroups), or bilateral synchronous communication (telephone). Several attempts have been made to integrate multilateral synchronous communication in ODL. Most of them have the disadvantage of expensive equipment and high bandwidth requirements. This is why typical seminars/workshops at University of Hagen are still on-site events. Seminars and workshops are among the last 'presence' remains at the distance-teaching University of Hagen. That is, students have to travel to the university and being there physically at the time of the seminar. Within this article, the term 'seminar' will be used both for seminars and workshops.

2. TECHNICAL BACKGROUND

The approach presented is based on low-cost PC hardware, low-bandwidth communication channels, and open-source/free software. It combines text chat, voice-over-IP, application sharing, and a virtual reality environment. The main means of communication is multipoint voice-over-IP. It is the most natural way of communication, enabling users to discuss with (almost) no technical barriers and to react very quickly. If necessary, it can be supported by either broadcast or bilateral text messages. When working on a shared document, or resenting results to the other participants, the respective program can be operated by the participants. The documents are stored on a central server, which can be controlled remotely. All participants can see the results of the operations immediately. An additional way of communication is via a virtual reality environment. Users can virtually meet in a (real) room, seeing avatars of the others. They can express moods or wishes by a set of actions of the avatars. The application sharing can be integrated in this room, e.g. by projection on a (virtual) screen. The virtual reality environment gives the feeling of a 'real' collaboration, whilst saving bandwidth.

3. PREREQUISITES

To avoid the effort of time and money for the students a synchronous communication feature is highly desirable. As a future perspective of a distance teaching university, even a development towards a 'virtual university' is thinkable, which should be completely internet-based.

Asynchronous virtual seminars started at the University of Hagen about 20 years ago. But asynchronous events have several disadvantages regarding collaboration in a group. To provide a synchronous seminar-like environment to a group of students, typical synchronous communication techniques like video-conferencing are the first choice. Unfortunately, they require high bandwidth capacities. A video-conference with more than two communication partners is a typical point-to-multipoint application. If a true collaboration of all partners is desired, the partner with the smallest bandwidth limits the communication.

For synchronous seminar-like events, a real interaction both between the students and between students and the tutor is required, so a bandwidth-saving way of interaction is needed as an alternative to the video based communication. Pure text-based communication (chat) does not meet the requirement of real interaction. In a 3D-chat an *avatar* tries to mimic the behaviour of the user in virtual reality. An *avatar* is the representation of a real person in virtual reality. The *avatar* substitutes the video picture in a video conference. Participants can see each other as in a real world scenario. It is possible to provide the user avatars with some simple gestures to visualize emotions, e.g. 'agreement' or 'disagreement'. Bandwidth-consuming loading of the virtual environment is done once at the beginning of the event. After that, only changes are submitted. Therefore, bandwidth requirements for 3D-chats are much lower than that of video conferences.

4. VIRTUAL ENVIRONMENT

Participants of a seminar should get the impression of meeting each other in a common room. Therefore, a well known seminar room of University of Hagen has been modelled in VRML (Virtual Reality Modelling Language). On the client side, VRML is used to display the virtual 3D environment. VRML as a text-based language is a powerful, nevertheless simple language to build virtual worlds. These include 3D objects, light sources, and animations. This virtual room is equipped with a virtual beamer and screen, to render a live audio/video stream (e.g. a lecture) or the desktop of a shared computer. The introduced multi-user virtual reality seminar environment consists of

- a basic text chat, which is used if the other means of communication fail,
- a virtual seminar room with user avatars,
- a voice-over-IP audio connection, and
- a shared virtual computer for 'beaming' of presentations to the clients.

5. COMMUNICATION MIDDLEWARE

The communication middleware is based on the open source VRML multi-user software DeepMatrix [1], which implements its functionality by Java-VRML coupling. DeepMatrix offers an educational licence that is royalty free. VRML specifies an External Authoring Interface (EAI) which can be used by external Java-based applications to monitor and control the VRML environment. This is used to update the virtual world with the positions of the other user avatars. DeepMatrix itself is a pair of client and server software implemented in Java [2]. The server is implemented as a Java application which communicates with all clients and provides them with updates of the 3D-scene. The client applet controls the local VRML browser-plug-in via the EAI to update the scene (i.e. the positions of other avatars) and senses the local user movements to send new positions to the server.

An additional DeepMatrix client, based on Eyematics Shout3d [3] VRML rendering engine, provides an own API for Java-VRML coupling. Since the Shout3d applet is pure Java based, some drawbacks like slower 3D-rendering (the Java Virtual Machine is not allowed to use the clients graphics hardware acceleration), and the implementation of a subset of the VRML standard only must be accepted. The most important advantages of the DeepMatrix-Shout3d client are: no VRML-browser binary is necessary, and no additional software need to be installed on the client computer. Therefore, this solution is platform independent, both with respect to hardware as to software. The original Geometrek implementation of the Shout3d based clients does not allow any interaction between users of the VRML-browser based and Shout3d based clients. We modified the Shout3d based clients to provide shared rooms for users of both clients. Due to the limitation of the Shout3d VRML implementation avatar gestures and streaming video rendering inside the 3D-window are not realized.

6. USER AVATARS

All remote users are represented by realistic human avatars. Every user is able to control gestures of his avatar. Some of these avatar-gestures are especially adapted to a typical classroom situation, e.g. ‘put one’s hand up’ or ‘point to’, to provide non-verbal communication to the users. This virtual environment is equipped with a virtual beamer and screen, to render a live audio/video stream (e.g. a lecture) or the desktop of a shared computer (figure 1).

Fig. 1: Virtual seminar room with avatars

7. AUDIO CONFERENCING

Audio is the most important means of communication in seminars. To enable usage of only one connection line for the client computers, we decided to use voice-over-IP with the open source, MBone based RAT audio tool from UCL [4]. We are using RAT in unicast-mode with a unicast multipoint reflector. RAT supports the UDP-based H261 protocol and GSW-codec, which allows real time audio with a typical bandwidth of 10kbit/s for a single speaker.

8. STREAMING VIDEO

Due to the relatively high bandwidth requirements, the streaming video feature is only used if bandwidth capacity is sufficient for all participants. The combination of a virtual but nevertheless realistic environment with user avatars pretends the users of meeting in a common room and seeing each other. Previous developments with on-line, remote experimentation in a real laboratory verified the usability of ‘faked’ reality [5,6,7].

9. A SHARED VIRTUAL COMPUTER

To provide the users with a kind f a whiteboard a universal solution was selected. Not only presentation software like PowerPoint is frequently used to explain topics to a group of users. In different disciplines user-groups need different software tools during seminar events. A universal solution is a shared PC, simultaneously useable to all participants. A platform independent solution for simultaneously remote control of computers is the open source 'Virtual Network Computing' (VNC) tool developed by the AT&T Laboratories in Cambridge [8]. The HTTP-server which is part of the VNC server utility is running on the shared seminar computer, and provides a Java applet to the client computers, which displays the desktop content of the shared computer in a web browser window. All the connected and authenticated users are able to remote control all installed applications. Documents and binaries can be uploaded easily to the shared computer via standard e-mail attachments. The screen resolution and colour depth should be reduced to 640x480x8 due to bandwidth limitations of remote modem users. A screenshot from a participant's PC with audio conferencing, whiteboard and virtual seminar room is shown in figure 2.

10. IMPLEMENTATION AND TEST

The whole system was implemented and tested at University of Hagen in May 2004. A server at the university acted as main hub for a seminar. It hosted a chat application, an audio reflector, and a virtual reality server. In addition, this server hosted the shared applications (Acrobat and PowerPoint) which were remotely controlled by the participants. After a lot of tests in bilateral and multilateral collaboration sessions, a final seminar took place with a total of 12 participants from all over Germany. Some of them had modem-line connections (56kbit/s) only. The system worked very reliable and offered high quality communication to all participants.

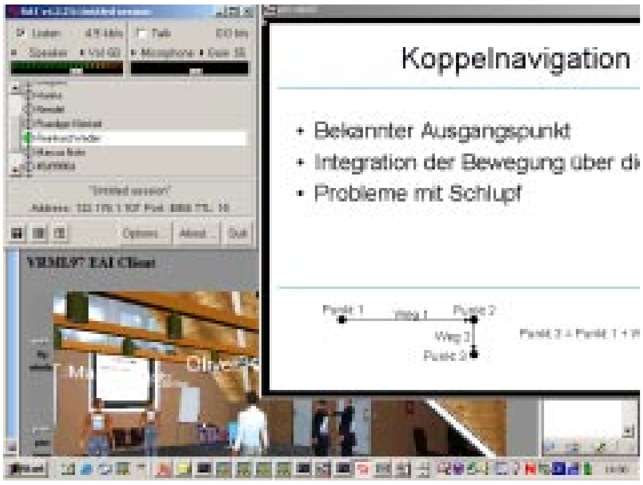

Fig. 2: Screenshot with conferencing tools

REFERENCES

[1] Geometrek: DeepMatrix. http://www.geometrek.com/products/deepmatrix.html

[2] Sun Microsystems: Java. http://java.sun.com

[3] Eyematic Interfaces Inc.: Shout3d. http://www.eyematic.com/products_shout3d.html

[4] University College London: RAT. http://www-mice.cs.ucl.ac.uk/multimedia/software/rat/

[5] A. Bischoff, C. Röhrig: A Multiuser Environment for Remote Experimentation in Control Education. Proceedings IFAC Workshop on Internet Based Control Education. Madrid, Spain 2001.

[6] A. Bischoff, C. Röhrig: Streaming Audio/Video and Multiuser Virtual Reality based Environment for Collaborative Remote Experimentation. Proceedings 21st ICDE World Conference on Open Learning and Distance Education, Hong Kong 2004.

[7] A. Bischoff: Web Based Collaboration System for Workshop-Like Events. Proceedings IADIS International Conference, Web Based Communities. Lisbon, Portugal 2004.

[8] RealVNC Ltd: http://www.realvnc.com/index.html