

# Streaming Audio/Video and Multiuser Virtual Reality based Environment for Collaborative Remote Experimentation

Andreas Bischoff and Christof Röhrig,  
 Department of Electrical Engineering,  
 and Information Technology,  
 University of Hagen,  
 D-58084 Hagen, Germany

*Abstract*—

**This paper presents a platform independent approach to a multiuser virtual reality remote experimentation system. It addresses a wide area of problems occurring in conjunction with remote laboratory experiments, starting with access management procedures via remote control to network-based analysis of measured data. Also distributed multiuser related problems like interaction and shared resources are covered. The collaborative environment introduced here allows experimentation in a team. The group is able to interact and to discuss the results of their work. A real collaboration like in local experimentation is possible. Students have access to the experiments via Internet from anywhere at any time. They control the experiments exclusively with their standard Web browser, no additional software such as analyser tools are needed. The remote laboratory is based on a client/server architecture, which is mainly implemented in the Java programming language. Also all components are based on standards like VRML and SQL. The methods and software modules developed for the lab are generic and frequently used in several remote experiments at some German universities.**

## I. INTRODUCTION

The University of Hagen is the first and only university in German-speaking countries, which is (almost) exclusively based on distance teaching methods. As one of the largest universities in Germany it provides university-level education and related degrees. About 80% of the students are already professionals who study mainly in the evening and on weekends. The Internet becomes increasingly important as a medium for knowledge distribution and even as a learning environment.

In distance teaching, laboratory experimentation is inconvenient because the students usually have to be physically present in the universities' labs.

One solution to avoid this disadvantage is virtual experimentation. In this paradigm the experiments are simulated and visualized by means of virtual reality. Another concept is remote control of laboratory experiments. Early implementations are reported in [1] and [3]. Providing remotely accessible experiments, unique or expensive equipment can be shared between different universities. So, a larger number of laboratory resources is available, and students can choose from a variety of lab experiments.

In local laboratory experimentation students usually work together in groups of two or more. This learning paradigm is often called collaborative learning. Collaborative learning develops skills for solving problems in a team. The underlying premise of collaborative learning is based upon consensus building through cooperation by group members, in contrast to competition in which individuals rival to other group members. Members of the learning group will usually organize their activities themselves and decide upon the roles of the different members via consultation and negotiation.

With the rapid expansion and availability of communication and information technologies, collaborative learning can also be done effectively in a virtual environment at a distance. Collaborative virtual environments bring together users, which are geographically distributed, but connected via

a network.

The remote laboratory at the University of Hagen is used for teaching of 'control theory'. The experiments use simple controllers with fixed structure. Students select the controller algorithm and define controller parameters remotely. There are some slightly different approaches reported, where students upload the controller algorithm to the experiment or where the controller algorithm is optionally executed on the client computer. The described experiment plant itself is a mobile robot platform.

#### A. Video and Audio Stream

A live video stream for viewing the experiment and an optional audio stream helps to provide a laboratory feeling. The special requirements for online experimentation restrict live video streaming technologies to methods with no or minimal delay. The Java Media Framework [10] was chosen to display the media streams in the browser. JMF is an API for incorporating media data types into Java applications and applets. The JMF API is a cross-platform solution written entirely in the Java programming language. Just one single applet is needed to receive the video and audio stream from the server. This applet is stored on the server eliminating the need to install it at the client. The user is able to choose different transfer rates for the video-stream to meet individual bandwidth limitations. The video-stream itself is based on the Internet-UDP-protocol (User Datagram Protocol, RFC 768). The UDP is very usable for streaming applications, because it's performance (UDP does not provide sequencing of the packets in which data arrives in). Details of the implementation are described in [9]. Well known techniques like Realnetworks [8] and Microsoft Windows Media streaming [4] does not meet the requirements because a very high compression is bought dearly with a streaming delay of usually 5 to 10 seconds. The video quality of these technologies is much higher than the quality of a minimal delayed JMF video stream ( $\leq 1$  second) because the compression algorithms are able to "look into the future" to avoid the retransmission of pixels which are changing only for a short period of time. High bandwidth consuming streaming methods like server-push [5] are not usable

because most of our students still using analog modem-connections (56kbit/s).

#### B. Collaborative environment

The collaborative environment is divided into two main modules: a rendering and graphics part on the client side and a communication middleware on the server side. On the client side, VRML is used to display the virtual 3D environment. VRML specifies an External Authoring Interface (EAI) which can be used by external applications to monitor and control the VRML environment. This is used to update the virtual world with data of the experiment and positions of the other user avatars. User interfaces via comfortable Java applets can be build in order to give access to the VRML environment and to allow higher-level modifications. The communication middleware is based on the open-source VRML-Multi-User-Software VNET which implements its functionality by Java-VRML coupling via the EAI. VNET itself is a pair of client and server software implemented in Java [11]. 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-plugin via the EAI to update the scene (the positions of other avatars) and senses the local user movements to send new positions to the server. The communication protocol between client and server is well documented, so it is possible to interface the VNET multi-user software with the teleoperated laboratory server-system. The lab server acts as an additional client of the VNET server. This additional client module provides the VNET server with the position and orientation changes during the teleoperated experiment. The remote user views the virtual representation (the avatar) of the experiment (a vehicle) in her/his VRML-browser window. All the 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' and 'point to', to provide non-verbal communication to the users. The user-avatars are generated by Avatarstudio [2] and are modified to match VNET's avatar EAI-event specifications.

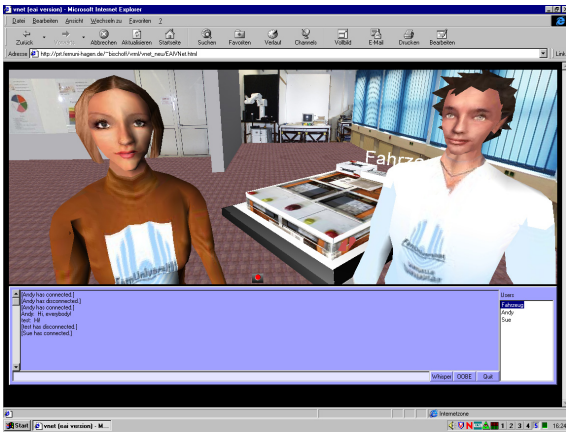


Fig. 1. 3D-Chat Interface

An alternative way of displaying 3D-information of user-avatars is a combination of video and 3D-data, the so called augmented reality. In this case a VRML-representation of all the present avatars in a corresponding view to the real camera-system is generated on server-side and mixed into the live-videostream by a genlock device (a video mixing device). All those virtual and video representations are provided to give the user the impression of being physically present at the experiment, and to allow real collaboration of a group of students.

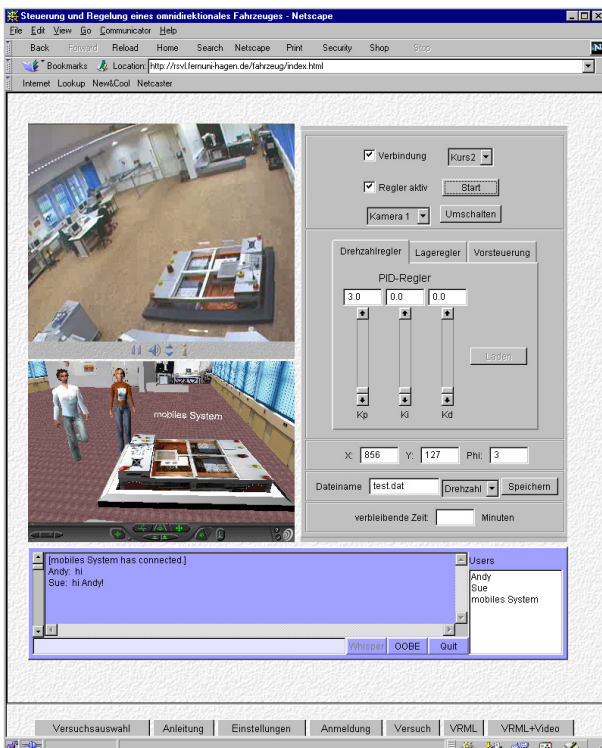


Fig. 2. Web Browser with JMF, Virtual Reality 3D-Chat and Control Applet

### C. Displaying of Data

While the experiment is in progress, the input and output data of the experimentation plant are measured and stored as an ASCII file on the server. The file is platform independent and includes the input and output values of the plant and controller. Students download the file for later analysis or they analyse the data online with the analyser applet. The analyser applet creates graphs from measured data and displays them in analysis sheets. The analyser provides user selected colors, automatic or manual scaling and rastering.

Analysis sheets can be directed to a printer or stored into a file. This is a useful feature for the students' final elaboration of experimental results.

Fig. 3 shows the analyser windows.

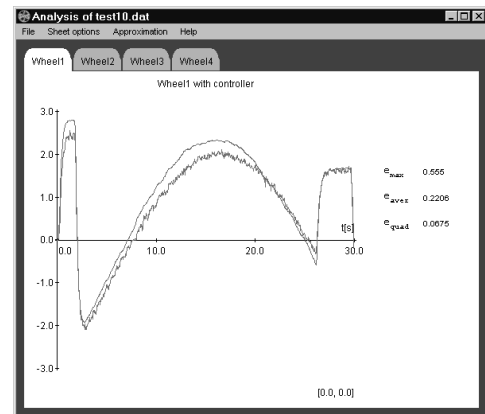


Fig. 3. Analysis of the Output Data

### D. Firewall limitations

The experiment control applet itself and the VNET-virtual reality collaboration applet are opening socket connections to the server application. A user behind a firewall must contact the network administrator to open the required ports. An alternative way to avoid problems with firewalls is the tunneling of connections through a VPN<sup>1</sup>. The data processing service centre of the university of Hagen offers a VPN-service based on IPSec[7], which allows all students to tunnel all TCP/IP traffic through a VPN-client. Another benefit of the usage of VPN is that the users get a routable IP address inside the university IP subnet, which avoids problems with limited internet

<sup>1</sup>Virtual Private Network

access via NAT (Network Address Translation).

## II. CONCLUSION

The contribution shows that distance education methods can be applied to real laboratory experiments. Even collaboration in a team is possible. Remote laboratory users interact and discuss the results of their work. A real collaboration like in local experimentation is possible. In the opinion of the authors such functionality justifies the usage of avatars in education. On the client side, there are only some minor requirements. Students are able to use the experiments with a Web browser and Java runtime environment. In the laboratory, the experiment must be adapted to the requirements of remote control. The server exclusively uses software that is available for free. All software developed for the lab has been implemented in the programming language Java. Therefore, it can be easily adapted to different platforms. All the components are usable through a low bandwidth modem (56kbit/s) or ISDN-connection. The remote experimentation without the live-video-stream, but including the virtual reality collaborative environment, is still usable under 'worst case' conditions, like a 9600 Baud GSM-mobile-phone-connection.

## III. FUTURE WORK

The university of Hagen and German partner universities are developing other components for remote control laboratories. This work will be done for the ongoing German federal research project 'learnnet' (LEARNing and experimentation in the NET)[6] components for remote control laboratories. Especially components for video-streaming, access managing and user authentication will be reused and enhanced.

## REFERENCES

- [1] B. Aktan, C. Bohus, L. Crowl, and M. Shor. Distance Learning Applied to Control Engineering Laboratories. *IEEE Transactions on Education*, 39(3):320–326, 1996.
- [2] Blaxxun, Canal+. Avatarstudio. <http://www.avatarstudio.com>, 2001.
- [3] J. Henry. Controls Laboratory Teaching via the World Wide Web. In *Proceedings of the ASEE Annual Conference*, Washington, USA, 1996.
- [4] Microsoft. Windows Media. <http://www.microsoft.com/windows/windowsmedia/>, 2000.
- [5] Netscape. Server push. <http://www.netscape.com>, 1997.
- [6] N.N. Learnnet. <http://www.learnnet.de>, 2001.
- [7] N.N. Ipsec. <http://www.ietf.org/html.charters/ipsec-charter.html>, 2002.
- [8] Realnetworks. Real-Server, Real-Producer, Real-Player. <http://www.realnetworks.com>, 1998.
- [9] C. Röhrig and A. Jochheim. The Virtual Lab for Controlling Real Experiments via Internet. In *Proceedings of the 11th IEEE International Symposium on Computer-Aided Control System Design*, pages 279–284, Hawaii, USA, Aug. 1999.
- [10] Sun Microsystems. Java Media Framework. <http://www.java.sun.com/products/java-media/jmf/>, 2000.
- [11] S. White and J. Sonstein. VNet. <http://www.csclub.uwaterloo.ca/~sfwhite/vnet/>, 1998.