**Proceedings of the 42nd IEEE
Conference on Decision and Control
Maui, Hawaii USA, December 2003**

Web-based Environment for Collaborative Remote Experimentation

Christof Röhrig and Andreas Bischoff,
Department of Electrical Engineering
University of Hagen
D-58084 Hagen, Germany
Email: christof.roehrig@fernuni-hagen.de

***Abstract*— Remote experimentation helps to overcome restrictions in time and space. In local experimentation students usually work together in groups of two or more. This learning paradigm is often called collaborative learning. This paper presents a Web-based approach to collaborative remote experimentation. 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 is needed. The remote laboratory is based on a client/server architecture, which is mainly implemented in the Java programming language. The methods and software modules presented in this paper are generic and frequently used in several remote experiments.**

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 [1]. Simulation is a proper way to complement engineering education but in general it cannot replace experiments on real plants. Since simulation is only as good as the model, experimentation has the advantage of making the user aware of phenomena that are hard or impossible to simulate.

Another concept to avoid the disadvantages of local experimentation is teleoperation of laboratory experiments. Early implementations are reported in [2] and [3]. An overview of recent developments in control education is given in [4]. Providing teleoperated experiments, unique or expensive equipment can be shared between several universities. So, a larger number of laboratory resources are 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 best other group members. Learning members of the group will usually organize their activities themselves and decide upon the roles of the different members via consultation and negotiation [5].

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.

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.

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 [6] or where the controller algorithm is optionally executed on the client computer[7]. Another different approach to a Internet-based Lab using NetMeeting is given in [8]. A collaborative environment for research on a motor experimentation platform is described in [9]. Java-based tools for multimedia collaborative environments are presented in[10].

II. System Architecture

The technology behind any synchronous collaboration tool is a mechanism that enables a user to send updates to other users about the interactions that are made in the shared environment. It is necessary for the participants to have the same view of the application in real time [11]. A comparative summary of the different standards for collaborative environments is given in [12].

The main design idea of the collaborative remote experimentation system is to use the World Wide Web as communication structure and a Web browser as user inter-

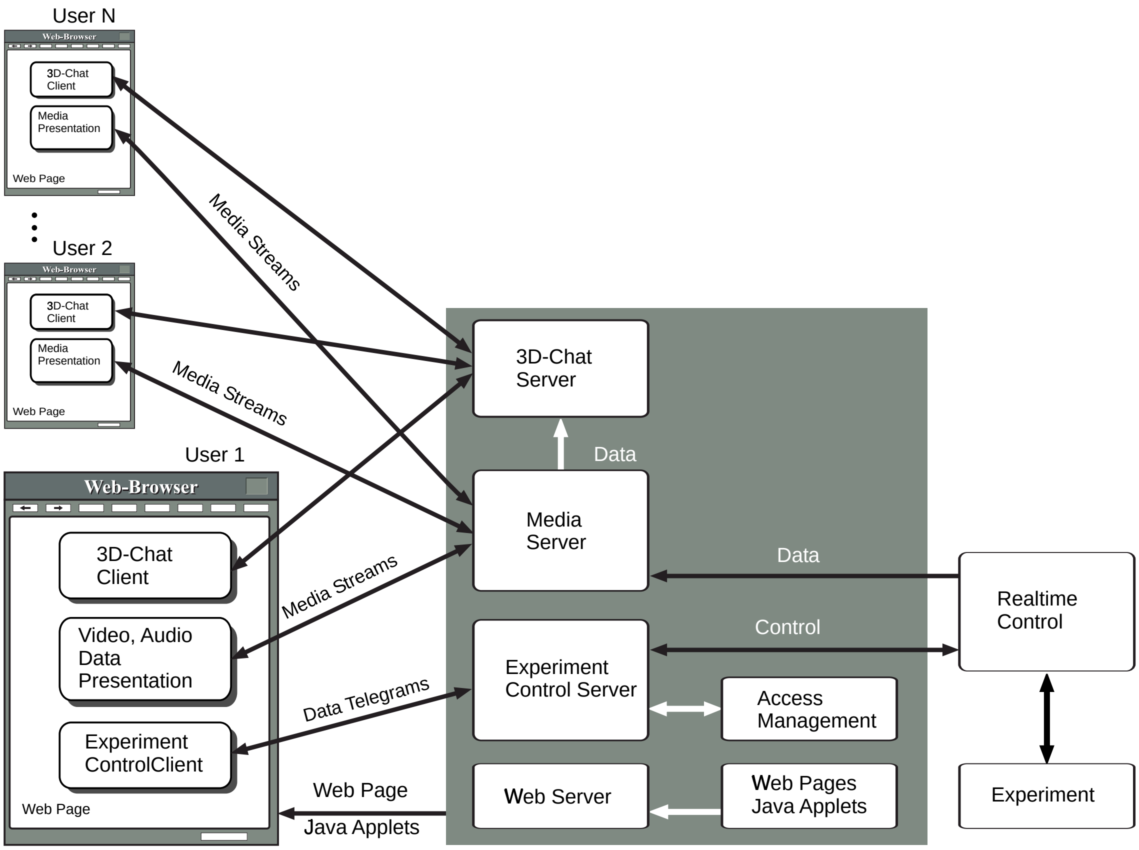

Fig. 1. Communication Structure

face. The Web browser provides a platform for transmitting information as well as an environment to run the client software. A Web server is the interface between the client and the experiment. The Web itself provides the infrastructure to exchange the necessary information. Fig. 1 shows the communication structure of the remote lab.

The communication structure is based on a client/server architecture written mainly in the Java programming language. Students may work on any platform that supports a Web browser with a Java runtime environment. Java is used to eliminate the operating system problem of heterogeneous environments, such that users are not restricted in their choice of a resource. This is specially important for distance education since some users might choose UNIX-workstations, while others might prefer Windows 95/98/NT PCs or MACs. The introduction of Java helped to overcome these problems. The local Web browser is the only user interface to the experiment. The browser loads the client software as Java applets from the server and starts them. Due to the modular structure of the system, extensions with new features are easy to implement. The Web server provides the Web-pages, the VRML scenes, VRML avatars and all Java applets. The server hardware includes a video capture card and a sound card for video and audio grabbing. The real-time controller of the experiment is usually implemented on a different computer hardware with a real-time operating system. The communication structure is generic. Therefore, it can be used for different experiments. Java applets on the client's side allow the continuous improvement of the software since the applets are loaded when they are needed. Applets are always up-to-date so no user software upgrading is necessary when the software is exchanged.

A. 3D-Chat Environment

The 3D-chat 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 [13]. 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 experimentation 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 [14] and are modified to match VNet's avatar EAI-event specifications.

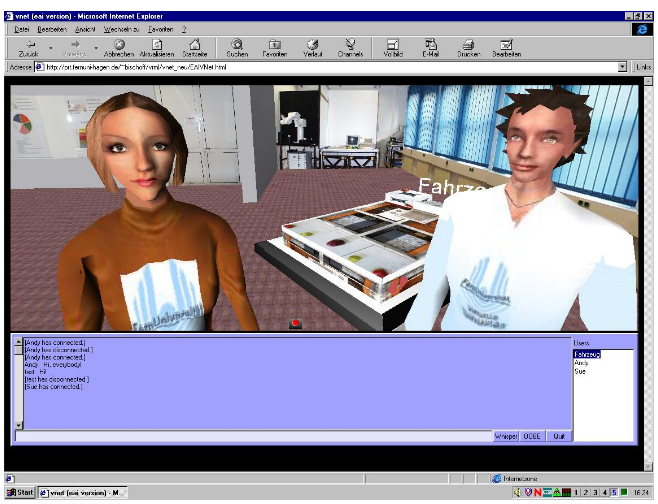

Fig. 2. 3D-Chat Interface

B. Video, Audio and Data Streaming

A live video stream for viewing the experiment and an optional audio stream helps to provide a laboratory feeling. The interaction of the users with the experiment restricts live video streaming technologies to methods with minimal delay. Well know techniques like Realnetworks [15] and Microsoft Windows Media streaming [16] does not meet the requirements because a very high compression is bought dearly with a streaming delay of usually 5 to 10 seconds. This technologies are designed for broadcasting of films or live events and not for highly interactive applications. High bandwidth consuming streaming methods like server-push [17] are not usable because most of the students at home are still using analog modem-connections (56kbit/s).

The Java Media Framework (JMF) [18] 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. The user is able to choose different transfer rates for the video-stream to meet individual bandwidth limitations. Details of the implementation are described in [19].

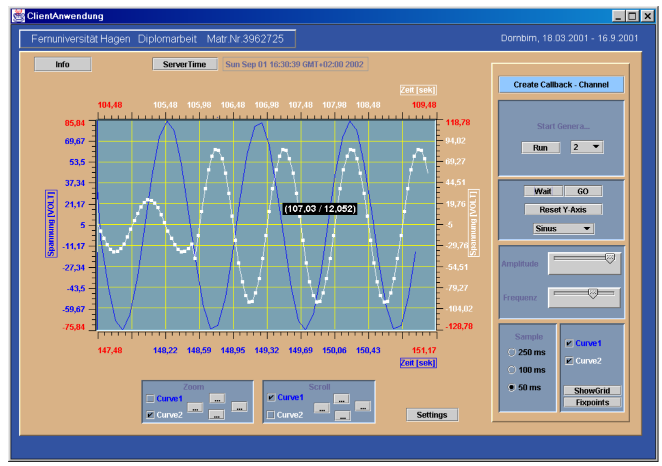

Fig. 3. Data Streaming

While the experiment is in progress, the input and output data of the experimentation plant is measured and streamed in real-time to the clients. The data are platform independent and include the input and output values of the plant and controller. An analyzer applet receives the data stream and creates graphs from measured data and displays them in analysis sheets. Fig. 3 shows the analyzer applet. The analyzer 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. Students can save the date in a file for later analysis or they analyze the data online with the analyzer applet [20].

C. Access Management

Scheduling of experimentation time is performed by an access management system. Appointments are stored in a SQL database. The user interface of the access management system is implemented in Java and executed in the Java runtime environment of a Web browser. It is subdivided into two parts, an administrator interface and a student interface. The administrator interface consists of several dialog boxes for creating and deleting accounts, for setting up individual time-quotas, for defining time slots and for analyzing the

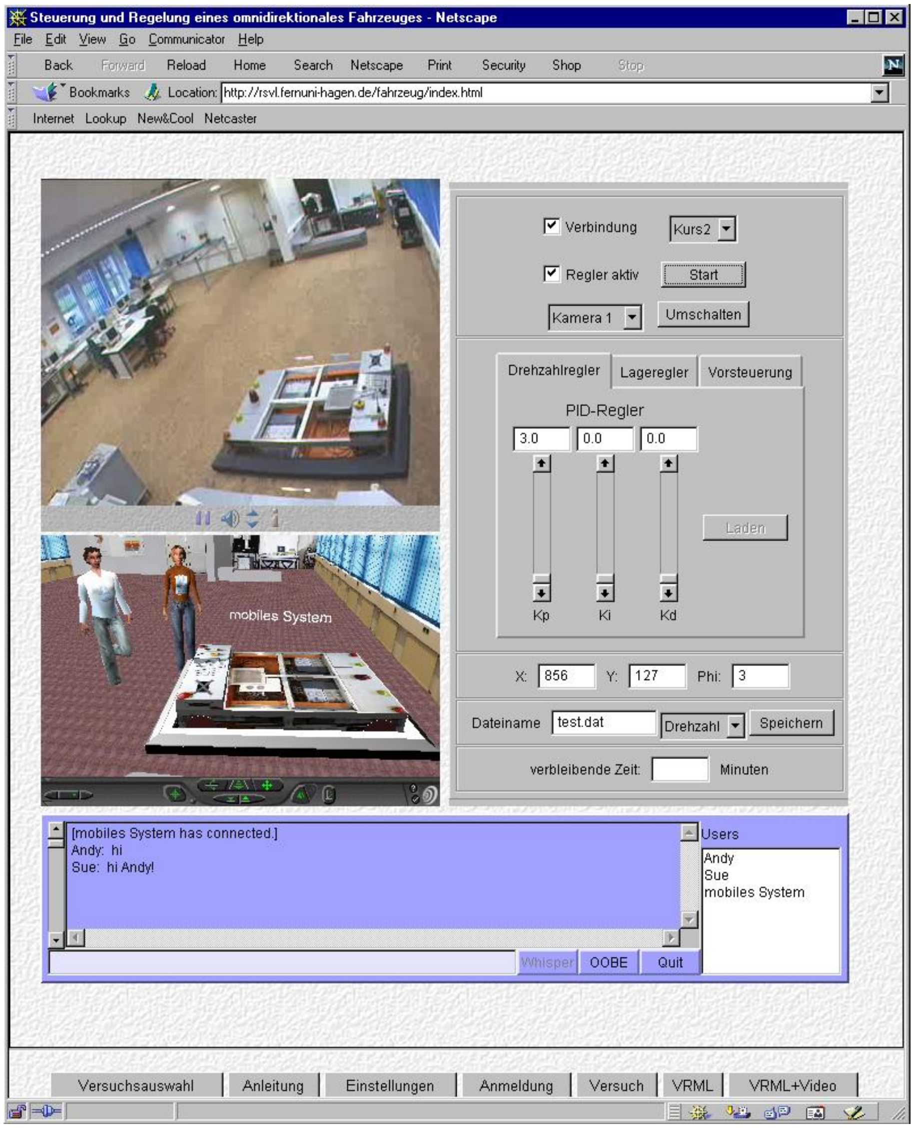

Fig. 4. Web Browser with JMF, Virtual Reality 3D-Chat and Control Applet

logging messages. More details of the access management system are described in [21].

D. Experimentation Interface

The local Web browser is the only user interface to the experiment. The browser loads the client software as Java applets from the server and starts them. Due to the modular structure of the system, extensions with new features are easy to implement. Three Java applets are included in the Web-page. One applet receives the live video and audio stream of the laboratory and another applet controls the experiment. The third applet implements the Virtual Reality 3D-Chat. Fig. 4 shows the Web browser with the video/audio Virtual Reality 3D-Chat and control applet. Details of the experiment itself are reported in [22].

III. Conclusion

The remote laboratory in online since summer 2000 [23]. Since then many students work successfully with our remote experiments. Distant students like to work remotely on experiments, because they save travel time and costs. The acceptance of remote experiments increases with better technical equipment on the students side. Nowadays most of our distant students use advanced communication techniques like ISDN or DSL. With this equipment the remote laboratory is very convenient to use. But also with low bandwidth modems all components are still usable. In this case the virtual reality environment offers high quality feedback of the experiment. In order to get a better laboratory feeling virtual reality should be complemented with high compressed video, even in the case of low bandwidth. Some students who just worked with virtual reality without any other feedback believed it was a simulation.

In the opinion of the authors, avatars help to get a feeling of collaboration. Students are very familiar with avatars, because of the popularity of computer games. The 3-D-Chat environment is a good supplement to the user interface and supports communication within the group and with the tutor. Students who have a job use the remote laboratory in the evening and on weekends. During this time it is difficult to organize tutor services. The 3-D-Chat environment helps to overcome smaller problems by group discussion.

Sharing interesting experiments with other universities in a remote laboratory saves time and resources. The university of Hagen and German partner universities are developing enhanced components for remote laboratories. This work is supported by the German Federal Ministry of Education and Research. In the project 'LearNet' [24] (LEARning and experimentation in the NET) re-usable components for media-streaming, data-analysis, access management and user authentication as well as an inverted pendulum/gantry crane experiment [25] will be developed.

IV. REFERENCES

[1] C. Schmid, "A Remote Laboratory Using Virtual Reality on the Web," *Simulation*, vol. 73, no. 1, pp. 13–21, 1999.

[2] B. Aktan, C. Bohus, L. Crowl, and M. Shor, "Distance Learning Applied to Control Engineering Laboratories," *IEEE Transactions on Education*, vol. 39, no. 3, pp. 320–326, 1996.

[3] J. Henry, "Controls Laboratory Teaching via the World Wide Web," in *Proceedings of the ASEE Annual Conference*, Washington, USA, 1996.

[4] S. Dormido, "Control Learning: Present and Future," *Proceedings of the 15th IFAC World Congress on Automatic Control*, July 2002.

[5] P. Kirscher, "Using Integrated Electronic Environments for Collaborative Teaching/Learning," *Learning and Instruction*, vol. 10, no. Supp/1, pp. 1–9, 2001.

[6] Y. Piguet and D. Gillet, "Java-based Remote Experimentation for Control Algorithms Prototyping," in *Proceedings of the 1999 American Control Conference*, vol. 2, San Diego, USA, 1999, pp. 1465–1469.

[7] J. Overstreet and A. Tzes, "An Internet-Based Real-Time Control Engineering Laboratory," *IEEE Control Systems Magazine*, vol. 19, no. 5, pp. 19–34, 1999.

[8] N. Swamy, O. Kuljaca, and F. Lewis, "Internet-Based Educational Control Systems Lab Using NetMeeting," *IEEE Transactions on Education*, vol. 45, no. 2, pp. 145–151, 2002.

[9] R. Sepe and N. Short, "Web-Based Virtual Engineering Laboratory (VE-LAB) for a Collaborative Experimentation on a Hybrid Electric Vehicle Starter/Alternator," *IEEE Transactions on Industry Applications*, vol. 36, no. 4, pp. 1143–1150, 2000.

[10] J. Oliveira, M. Hosseini, S. Shirmohammadi, F. Malric, S. Nourian, A. El Saddik, and N. Georganas, "Java Multimedia Telecollaboration," *IEEE Multimedia Magazine*, vol. 10, no. 3, pp. 18–29, 2003.

[11] S. Shirmohammadi, J. Oliveira, and N. Georganas, "Applet-Based Telecollaboration: A Network-centric Approach," *IEEE Multimedia*, vol. 5, no. 2, pp. 64–73, 1998.

[12] J. Oliveira, S. Shirmohammadi, and N. Georganas, "Distributed Virtual Environment Standards: A Performance Evaluation," in *Proceedings of the 3th IEEE/ACM International Workshop on Distributed Interactive Simulation and Real Time Applications*, Greenbelt, USA, Oct. 1999.

[13] S. White and J. Sonstein, "VNet," www.csclub.uwaterloo.ca/u/sfwhite/vnet/, 1998.

[14] Blaxxun, Canal+, "Avatarstudio," http://www.web3d-fr.com/Avatars/AvatarStudio/Download.php, 2001.

[15] Realnetworks, "Real-Server, Real-Producer, Real-Player," http://www.realnetworks.com, 1998.

[16] Microsoft, "Windows Media," http://www.microsoft.com/windows/windowsmedia/, 2000.

[17] Netscape, "Server push," http://www.netscape.com, 1997.

[18] Sun Microsystems, "Java Media Framework," http://www.java.sun.com/products/java-media/jmf/, 2000.

[19] 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*, Hawaii, USA, Aug. 1999, pp. 279–284.

[20] C. Sonnweber, "Design and Implementation of JavaBeans Library for Teleoperation of Laboratory Experiments," Diploma Thesis, University of Hagen, 2002.

[21] C. Röhrig and A. Jochheim, "Java-based Framework for Remote Access to Laboratory Experiments," in *Proceedings of the IFAC/IEEE Symposium on Advances in Control Education*, Gold Coast, Australia, Dec. 2000.

[22] A. Jochheim and C. Röhrig, "The Virtual Lab for Teleoperated Control of Real Experiments," in *Proceedings of the 38th IEEE Conference on Decision and Control*, vol. 1, Phoenix, USA, Dec. 1999, pp. 819–824.

[23] "Real Systems in the Virtual Lab," http://prt.fernuni-hagen.de/rsvl/info_e.html, 2000.

[24] "Learnet," http://www.learnet.de, 2001.

[25] H. Hoyer, M. Gerke, I. Masar, I. Ivanov, C. Röhrig, and A. Bischoff, "Virtual laboratory for Real-Time Control of Inverted Pendulum/Gantry Crane," in *11th Mediterranean Conference on Control and Automation*, June 2003.