

JAVA-BASED FRAMEWORK FOR REMOTE ACCESS TO LABORATORY EXPERIMENTS

Christof Röhrig,¹ Andreas Jochheim²

*Department of Electrical Engineering
University of Hagen
D-58084 Hagen, Germany*

Abstract: This paper presents a platform independent approach to remote experimentation. 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. It presents the architecture of the remote laboratory at the University of Hagen. 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 developed for the lab are generic and frequently used in several remote experiments at some German universities. Furthermore the paper discusses the motivation of remote experimentation.

Keywords: Laboratory education, Teleoperation, Control education, Educational aids, Multimedia

1. 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 which 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. As a future perspective of a distance teaching university, even a development towards a 'Virtual University' is thinkable, which should be completely Internet-based. Currently different technical approaches and learning methods are explored: courseware is offered via Internet including simulation interfaces, multimedia animations and on-line learn-

ers support. Seminars and examinations are held via video-conferencing, laboratory experiments become accessible worldwide.

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 visualised by means of virtual reality (Schmid, 1999). Another concept is remote control of laboratory experiments. Early implementations are reported in (Aktan *et al.*, 1996) and (Henry, 1996). 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

¹ email: christof.roehrig@fernuni-hagen.de

² email: andreas.jochheim@fernuni-hagen.de

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 (Piguet and Gillet, 1998) or where the controller algorithm is optionally executed on the client computer (Overstreet and Tzes, 1999).

2. SYSTEM DESIGN

The main design idea of the system is to use the World Wide Web as communication structure and a Web browser as user interface. 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.

2.1 Analysis of Requirements

Computer based controller implementation is necessary for remote experimentation.

Another important aspect is to transport the feeling of a real experiment to the remote user. A video and audio broadcast can provide the remote user with the feeling of being physically present at the location of the real experiment. With the visual feedback, the user supervises the experiment and checks whether the performance of the process is as expected.

For system identification purpose and controller evaluation, it is necessary to collect the relevant data of the process. These data have to be stored at the server for download and additional processing by the students. As only one student at a time receives access to an individual experiment, schedules and exclusive access procedures to the experiment are necessary. Users should be able to book laboratory time in advance. They should carry out the whole booking procedure by themselves in order to choose the time most appropriate to their needs.

The main requirement on the server side is the safety of the experiment and of the server computer. The experimental plant has to be protected against any action that can damage or destroy it. For this reason, all commands given to the plant controller must be analysed and dangerous controller parameters have to be avoided. If the controller algorithms can be defined by the user without any restrictions, system instabilities caused by the controller are hard to detect in advance. To evaluate students work, it is useful to log all communication between user and experiment.

From the users point of view there are some other important demands that have to be fulfilled before remote experiments become acceptable for them. Students do not have the choice of which computer to

use at their local site. They usually do not want to install specific software before they can start their experimentation. Therefore, cross-platform client software is an important user demand. Since students do not want to pay for additional software, it is necessary to provide client software for free.

2.2 Communication Structure

Fig.1 shows the communication structure of the system. 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. 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 and the Java applets. The server hardware includes a video capture card and a sound card for video and audio transmission. 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.

3. ACCESS MANAGEMENT

The laboratory administrator creates and deletes accounts, he defines time quotas for each experiment. After a user has requested a particular experiment, he downloads the instructions of the experiment. Some advance preparations of the experiment are necessary, which includes modelling of the plant and subsequent controller design. When the learner has finished all preparations, it has to be submitted to the professor in charge of that particular experiment. The responsible teacher checks the prepared work and -if successful- allows the student to make an on-line appointment for the experiment and to allocate a certain time-quota. So, in the next step access to a list of time slots is granted, which shows all dates available for experimentation. The appointment has to be made via an access management system for each experiment.

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 students interface (scheduler).

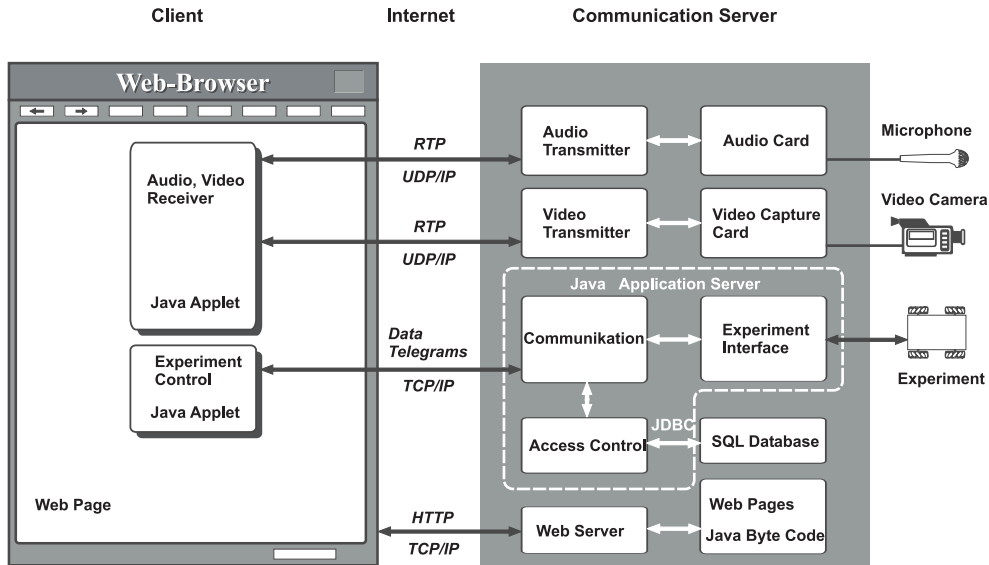


Fig. 1. Communication Structure

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 analysing the logging messages. The dialog boxes for administration are shown in Fig. 2.

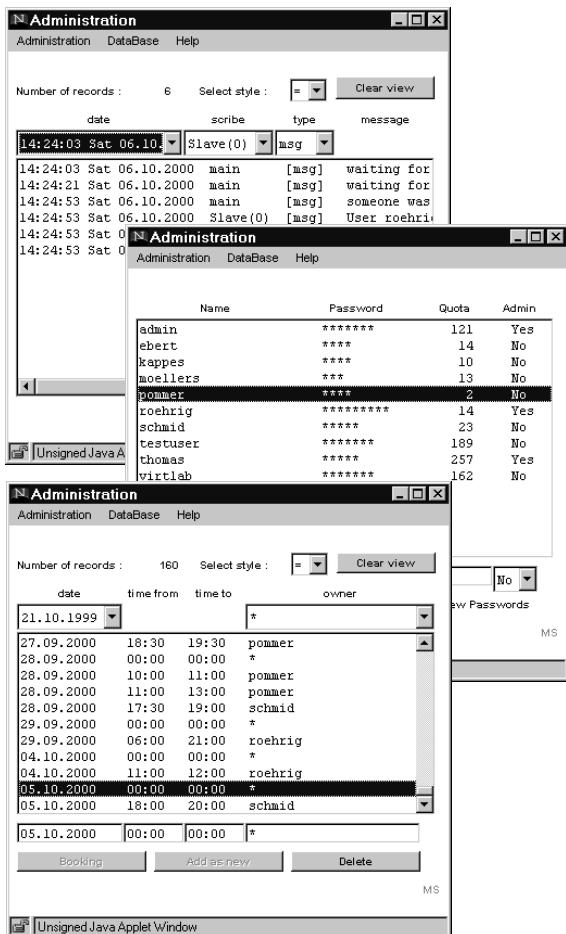


Fig. 2. Administration Dialogs

3.1 Scheduler

When students have received access to an experiment, they are able to book experimentation time. For logging into the server with the help of the Web browser, the student has to enter a valid user ID and password in order to pop up the scheduler dialog which is shown in Fig. 3. The user selects a week from a predefined list according to his needs.

Then the calendar shows the chosen week with free time slots indicated. The user can reserve the desired time directly with this calendar menu. Remaining times from the given time quota are monitored and available for future appointments. The actually booked appointment is stored in a database and is used for access control. At the booked time the access management system gives the user exclusive access to the experiment.

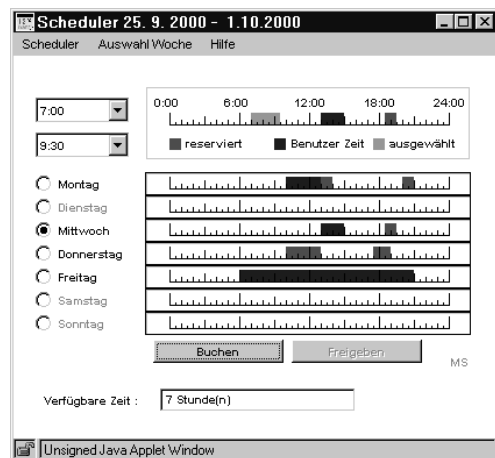


Fig. 3. Scheduler Window

3.2 Server and Database

The server is implemented as Java application. The communication between the server and the database is managed using the Java Database Connectivity (Sun Microsystems, 2000a). JDBC is an application programming interface (API) that enables access to database management systems (DBMS) from the Java programming language. It provides cross-DBMS connectivity to a wide range of Structured Query Language (SQL) databases. The SQL database mSQL (Hughes Technologies, 1999) is used with mSQL-JDBC (Imaginary, 1999). mSQL-JDBC is a database access API for the mSQL database engine that conforms to the Java JDBC API. The mSQL database and the mSQL-JDBC is freely available for non-commercial use. Fig. 4 shows the structure of the database. The database consist of three tables. This are the accounts table, the schedule table and the logging table. The user table stores user names, password, time quota and administrator flag. If the administrator flag is set, the user is administrator and manages the database. The logging table stores the log messages of the experiment. Messages include logging time and date, type of message and module name. Communication between client and server via Internet

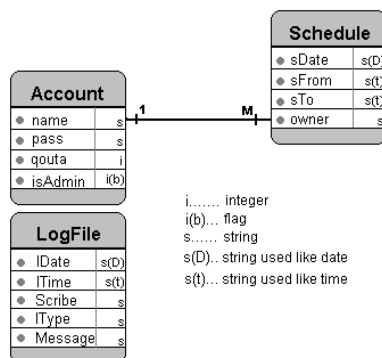


Fig. 4. Structure of Database

based on the ISO/OSI model and enhances the TCP/IP protocol. Messages between client and server are encrypted before they are sent via Internet. The encrypter uses the affine crypting algorithm with synchronised pseudorandom key generator (Straube, 1998).

4. EXPERIMENTATION INTERFACE

At the date of the experiment, the user connects to the server with a Web browser. The browser loads the Web page of the experiment. Two Java applets are included in the page. One applet receives the live video and audio stream of the laboratory and the other applet controls the experiment. Fig. 5 shows the Web browser with the video/audio and control applet. Details of the experiment itself are reported in (Jochheim and Röhrig, 1999).

4.1 Video and Audio Stream

A live video stream for viewing the experiment and an optional audio stream helps to provide a laboratory feeling. The Java Media Framework (Sun Microsystems, 2000b) 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. Details of the implementation are described in (Röhrig and Jochheim, 1999).

4.2 Experiment Control

The user interface for controlling the experiment is realized in Java. After the Web page is loaded, the applet starts and an authentication dialogue and a control window pop up. The user must enter a valid login name and password to access the server. When the login process is completed, the applet opens a TCP/IP connection to the access control module of the server.

The server reports the total time still available for the current experimentation. If no more time is left, the server cuts the students connection to the experiment.

The communication between client and server is executed as data telegrams using the open TCP/IP connection. These telegrams contain commands and optional parameters. The server acknowledges each command with a status telegram. This telegram contains status information of the experimentation plant which is shown in the Java applet on the client side.

The control applet and the plant interface depend on individual experiments and have to be adapted to each additional experiment.

5. DATA VISUALISATION AND ANALYSIS

For visualization of measured data, as well as support of the necessary system identification and controller design, a Java program was developed (Ondrejková, 1998). The program is executed online as applet in a browser or offline as an application in a Java Runtime Environment. The program is written in 100% pure Java. It requires a Java Runtime Environment (JRE) of 1.1.5 or higher. This JRE is included in common Web browsers (Netscape Communicator 4.08, Microsoft Internet Explorer 4).

5.1 System Identification

The first part of the experiment is to model the plant and to identify the parameters of the plant with the

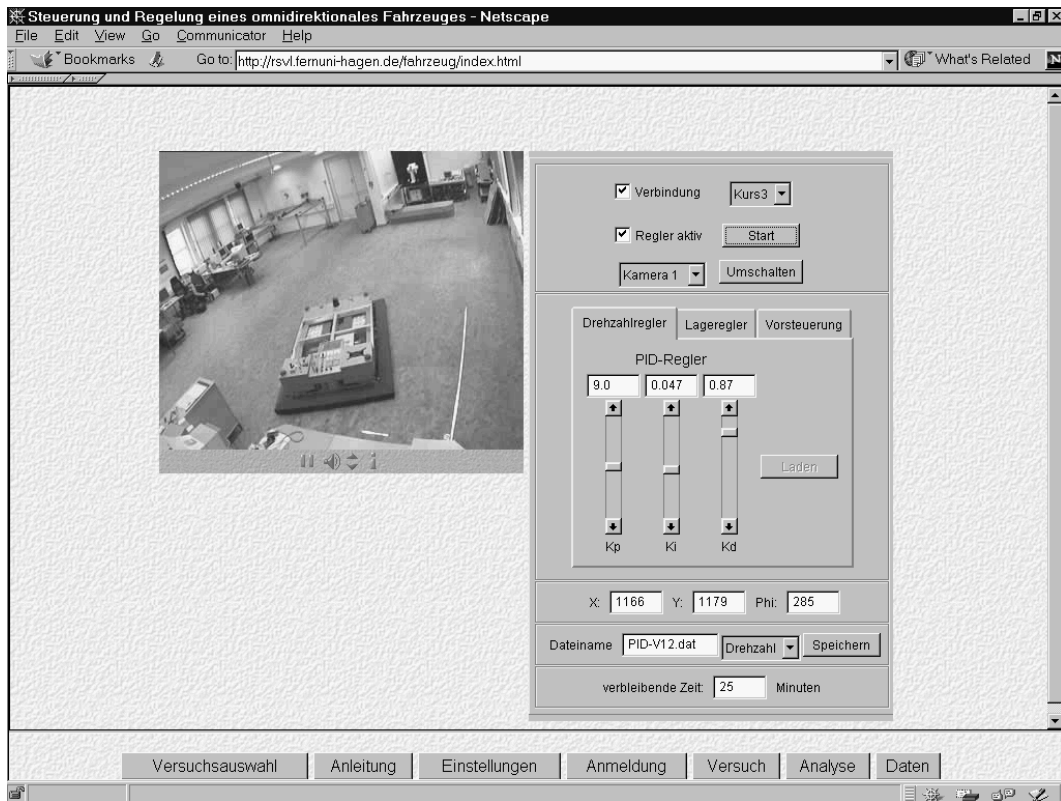


Fig. 5. Web Browser with JMF and Control Applet

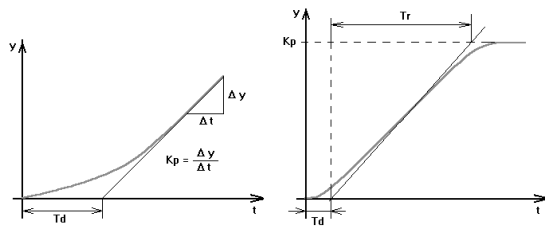


Fig. 6. Function Approximation

help of a step response signal. Measured values of the step response are stored on the server. Students analyse them with the analyser applet. The analyser applet identifies the model parameters of the plant by means of function approximation. For plants with integral behaviour it calculates the values *gain* K_p and *delay time* T_d , for those with non-integral behaviour *gain* K_p , *delay time* T_d and *rise time* T_r are calculated as shown in Fig. 6.

Fig. 7 shows the system identification dialog. In this dialog, the users set the rank and accuracy and start the iterative identification procedure. During the calculation, the dialog shows the actual accuracy. The approximated curve is shown after finishing calculation. The dialog displays parameters of time function and of transfer function.

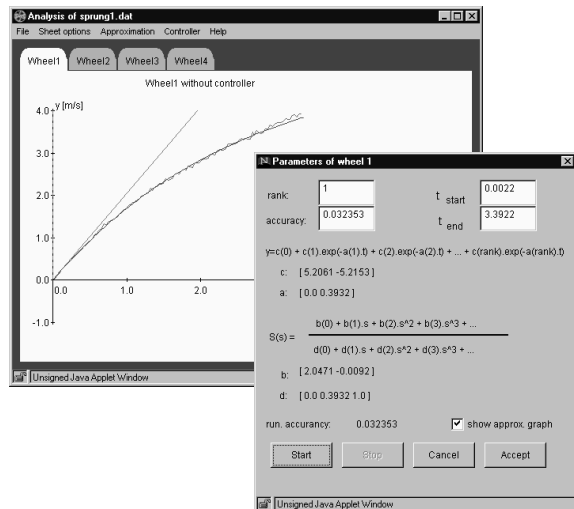


Fig. 7. System Identification

5.2 Controller Design

Another option of the analyser tool is the support of the controller design. In the controller design dialog, users set *delay time* T_d , *rise time* T_r and *gain* of the plant K_p and select the desired design method. Actually some automatic tuning methods (Ziegler-Nichols, Chien-Hrones-Reswick,...) are implemented.

5.3 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. 8 shows the analyser windows.

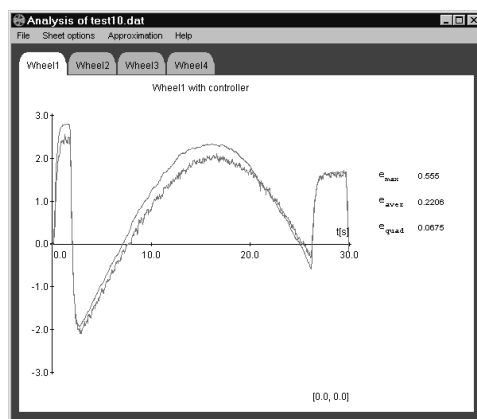


Fig. 8. Analysis of the Output Data

6. CONCLUSION

The paper shows that distance education can be applied to real laboratory experiments.

On the client (local computer of the student), 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.

To support teamwork, a multicast communication between any team of students is necessary. With the audio capturing and transmission features of JMF this can be done in a Java applet (Sun Microsystems 2000b). In this case, only one student at a time has active access to the experiment. At different stages of the experiment, other students can receive passive access to the experiment. These students observe the progress of the experiment, analyze system performance with different controllers, and discuss the results in order to modify the controller interactively.

7. REFERENCES

Project Website:

<http://prt.fernuni-hagen.de/virtlab/>

Aktan, B., C.A. Bohus, L.A. Crowl and M.H. Shor (1996). Distance Learning Applied to Control Engineering Laboratories. *IEEE Transactions on Education*, **39**(3), 320-326.

Henry, J. (1996). Controls Laboratory Teaching via the World Wide Web. In: *Proceedings of the ASEE Annual Conference*. Washington, USA, Session 3513

Hughes Technologies (1999). Mini SQL (mSQL). <http://www.Hughes.com.au/products/msql/>

The Center for Imaginary Environments (1999). mSQL-JDBC.

<http://www.imaginary.com/Java/Soul/>

Jochheim, A. and C. Röhrig (1999). The Virtual Lab for Teleoperated Control of Real Experiments. In: *Proceedings of the 38th IEEE Conference on Decision and Control*. Phoenix, USA, **1**, 819-824.

Ondrejková, R. (1998). Graphical User Interface to Evaluate Experiments in Virtual Lab. Diploma Thesis, University of Hagen, Slovak Technical University Bratislava.

Overstreet, J. W. and A. Tzes (1999). An Internet-Based Real-Time Control Engineering Laboratory, *IEEE Control Systems Magazine*, **19**(5), 19-34.

Piguet, Y. and D. Gillet (1999). Java-based Remote Experimentation for Control Algorithms Prototyping. In: *Proceedings of the 1999 American Control Conference*, San Diego, USA, **2**, 1465-1469.

Röhrig, C. and A. Jochheim (1999). 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, 279-284.

Schmid, C. (1999). A Remote Laboratory Using Virtual Reality on the Web, *Simulation*, **73**(1), 13-21.

Straube, M. (1998). Skeleton of Internet Application out of Deference Safeness. Diploma Thesis, University of Hagen, Slovak Technical University Bratislava.

Sun Microsystems (2000a). Java Database Connectivity. <http://www.java.sun.com/products/jdbc/>

Sun Microsystems (2000b). Java Media Framework. <http://www.java.sun.com/products/java-media/jmf/>