AC 2007-2564: AN ARCHITECTURE FOR REAL-TIME REMOTE LABORATORIES

Yaoye Li, Stevens Institute of Technology
Mr. Yoaye Li received his B.E. and M.S. degrees in Mechanical Manufacturing from Beihang University in 1996 and 1999, respectively. Currently, he is a pursuing a Ph.D. degree in Mechanical Engineering at Stevens Institute of Technology in Hoboken, New Jersey. His research interests include intelligent systems for design and manufacturing as well as remote experimentation.

Sven Esche, Stevens Institute of Technology
Dr. Sven K. Esche is currently holding a position as Associate Professor of Mechanical Engineering at Stevens Institute of Technology in Hoboken, New Jersey, USA. In 1989, he received an undergraduate degree in Applied Mechanics from Chemnitz University of Technology (Germany). After working for three years at Mercedes Benz AG in Stuttgart (Germany), he obtained M.S. and Ph.D. degrees in Mechanical Engineering from The Ohio State University in Columbus, Ohio, USA in 1994 and 1997, respectively. His current research interests include multi-scale modeling of thermo-mechanical processing of metals, integrated product and process design under conditions of uncertainty and risk as well as remote sensing and control of distributed devices with special focus on remote laboratories.

Constantin Chassapis, Stevens Institute of Technology
Dr. Constantin Chassapis is a Professor and the Director of the Mechanical Engineering Department at Stevens Institute of Technology. His research interests are in knowledge-based engineering systems; computer-aided design and manufacturing; structure-property modeling and characterization of polymers and polymer composites as well as in remotely controlled distributed systems. He has been an active member in ASME and SPE, and he has received a best paper award from SPE’s Injection Molding Division, the distinguished Assistant Professor Award at Stevens Institute of Technology, an Honorary Master’s Degree from Stevens Institute of Technology, and the Tau Beta Pi Academic Excellence Award.
An Architecture for Real-time Remote Laboratories

Yaoye Li, Graduate Student, YLi13@stevens.edu
Sven K. Esche, Associate Professor, SEsche@stevens.edu
Constantin Chassapis, Professor, CChassap@stevens.edu
Department of Mechanical Engineering, Stevens Institute of Technology, Hoboken, NJ 07030, USA

Abstract

The rapid spreading of broadband Internet access is enabling new methods of delivery for modern engineering and science curricula. This paper describes the design and implementation of a remote laboratory architecture, which allows the execution of experiments in real time. Contrary to previous remote-laboratory implementations based on a batch mode of operation, in this new system the students can interactively control the experiments and obtain the corresponding outputs (including raw data, data plots, video/audio streams and recordings) in an integrated browser-based user interface. Furthermore, this real-time system was designed in a platform-independent fashion such that it will facilitate its expansion beyond the boundaries of the original institution. A mechanical vibration setup is used to illustrate the system capabilities.

1. Introduction

Real experiments are indispensable in engineering and science education for developing skills for dealing with physical processes and instrumentation. The traditional way of conducting educational experiment is to go to a laboratory facility, where the experimental setups are located. There, the students typically work in groups (or sometimes individually) at a particular laboratory exercise and may receive tutorial help from instructors present at the site. During the last decade or so, Stevens Institute of Technology (SIT) as well as many other colleges and universities all around the world have begun to use Web-based remote laboratories. This trend of integrating such remote laboratories into undergraduate curricula is also supported by university administrators. A factor in support of this trend is the fact that physical experiments are expensive to maintain require dedicated facilities to accommodate the experimental setups. The rapid growth of the student populations in many institutions puts additional burdens on the spatial, temporal, financial and human resources required for operating traditional laboratories. On the other hand, due to the rapid increase available bandwidth of Internet, the materials that can be transferred on the network are no longer limited to static text accompanied by some static graphical images but can also include live or recorded audio and video content. The use of hypertext, hypermedia and multimedia in Web applications can potentially provide exciting and engaging learning environments, which will then lead to significant changes in the design of pedagogical processes and instructional approaches. Also, recently some evidence has started to be reported that Web-based remote laboratories can complement or even replace on-hand laboratories without compromising the desired learning outcomes.

The results of a recent study showed that the interface of remote laboratories based on the batch mode of operation fail to provide the students a feeling of actual presence. The students felt somewhat disconnected from the remotely located experimental apparatus using, yet they liked the general convenience provided by remote laboratories and the ability to carry out the
experimental procedures at their own schedule and pace. This paper presents some ideas that will contribute to improving the flexibility and interactivity of remote laboratories and thus alleviate some of the above mentioned concerns. These efforts are consistent with the attempts by many other institutions for enhancing the learning effectiveness through the application of the latest technologies. In this paper, the development and implementation of a real-time interactive remote laboratory are delineated. Several up-to-date streaming media technologies are employed in conjunction with various Web applications and protocols, including ASP.NET, TCP/IP, JavaScript, AJAX, and SQL Server. By using a multi-tier application architecture, it becomes possible to facilitate the continuous expansion of the current pool of experimental resources and to enable the flexible sharing of the experimental resources.

2. Related Work

Online laboratories can be classified into remote experiments (based on actual physical setups) and virtual experiments (based on pure software simulations). For instance, an interactive Web-based virtual laboratory on fluid mechanics was recently reported, which aims to enhance the students’ understanding of some complex concepts of fluid mechanics through simulations. This virtual laboratory integrates the analysis results with advanced graphics and animations. Thus, it attempts to provide the students with a realistic feel of the experiments.

On the other hand, remote laboratories consist of the actual hardware (i.e. experimental devices together with actuators and sensors to manipulate and monitor them remotely) and the appropriate software needed to perform experimental procedures. A batch-mode online semiconductor characterization laboratory was constructed using the shared iLab architecture, which introduces a piece of middleware (denoted as service broker) between a client application and a laboratory server. This middleware uses Web services to provide a generic set of functionality that is common to all laboratories. By employing this three-tier framework (i.e. client, service broker and laboratory server), this architecture expedites the development and simplifies the management of online laboratories. A real-time mechatronics/process control remote laboratory was developed to provide undergraduate and graduate students with a real-world, practical experience in modern DSP- and PC-based data acquisition and real-time control. Control panels based on Java applets and 3D-animations were designed to facilitate remote access to physical laboratory hardware. Telelabs, a cost-effective framework for providing an extendable set of batch-mode online laboratories, was recently presented. This system allows students to schedule the time they need to explore on their own the differences between theory and real equipment behavior. A corresponding study revealed that students using remote laboratories operated the experimental equipment for a much longer time than in conventional laboratory classes and as a result the learning outcomes seemed to be improved significantly. Another model for interactive virtual laboratory experiments based on Microsoft ConferenceXP learning infrastructure provides a real-time laboratory experience. With the embedded ConferenceXP tools such as online chat, presentation whiteboard, streaming video and audio, the students are enabled to work in collaborating groups on solving problems in real time.

The above mentioned models have different features and are implemented using different technologies. Building on these prior developments, a real-time remote laboratory system was designed, whereby special emphasis was placed on a high level of interactivity between student
and experiment. Rather than enabling “canned” experiments (i.e. fixed, predetermined experimental procedures), flexible experimental procedures can be conceived and implemented by the student. This approach facilitates explorative learning.

3. General System Architecture

![Figure 1: General system architecture](image)

The goal was to develop an architecture that will support the sharing of both categories of online laboratories, remote experiments and virtual experiments. This architecture represents the three-tier Web business architecture that dominates commercial applications. The three tiers of this architecture are as follows (see Figure 1):

1. The client represents the first tier, which consists of a PC or notebook with Internet connection. By means of a Java enabled standard Web browser, it can remotely access the virtual and remote experiment stations.

2. The middle tier is the resource manager, which consists of three servers to provide shared common services. The client communicates directly with the resource manager to carry out the experiment provided by the third tier. The services provided by the resource manager are as follows:
   - The Web server is responsible for accepting and responding to HTTP (Hypertext Transfer Protocol) requests from clients and for authenticating authorized users.
   - The database server is used for storing user records, experiment descriptions and experimental results.
   - The schedule server is used to generate experiment schedules and coordinate reservations for experimental stations. It is able to prevent conflicts and congestions.
3. The virtual and remote experiment stations represent the third tier. While the virtual experiment station has been described in detail elsewhere,\(^{15,16}\) this paper will mainly focus on the implementation of remote experiments. The remote experiment stations comprise the actual instruments themselves as well as a computer used for controlling and monitoring these instruments using actuators and sensors and for handling the communication with the resource manager.

4 System Implementation

4.1 Streaming Media

Today’s powerful computers with modern operating systems in conjunction with the latest advances in computer networking technology make streaming media practical and affordable for widespread usage. Streaming media are those that are consumed (heard or viewed) while being delivered. Live audio/video streaming improves the user’s feeling of physical presence at the experiment’s site. Therefore, the entire interaction between the user and the remote experiment is enhanced and the learning process becomes more effective.

There are three categories of currently existing commercial streaming media products that are readily applicable for distance learning.

- The streaming media framework allows a one-way streaming broadcast of real-time on-demand media or stored contents (e.g. Microsoft Windows Media\(^{17}\) and RealNetworks\(^{18}\)).
- The online conference and chat tools can support real-time interactive two-way or multi-way communications (e.g. Microsoft NetMeeting\(^{19}\), Windows Live Messenger\(^{20}\) and Yahoo Messenger\(^{21}\)).
- The real-time audio and video components can be used to develop specific communication requirements for one-way, two-way or multi-way communication functionalities. For example, “AXIS Media Control” offered by Axis\(^{22}\) can provide one-way real-time video, and “Voice Communicator” offered by Lakeofsoft\(^{23}\) can provide one-way, two-way and multi-way real-time audio.

It was first tried to implement real-time audio and video features for remote experiments based on the first two categories of streaming media products. It was found that the integration of these two product categories is very difficult and the high connection latency represents a serious issue. Thus, it was concluded that they are not so well suited for our purposes of implementing flexible interactive remote experiments. On the other hand, the streaming components of the third category do not only provide user-transparent audio compression/decompression for high-speed and high-quality audio/video transmissions as well as low latency for the processing of real-time media, but they can also be used to develop an integrated user interface for conducting remote experiments. Based on this comparison of the three categories, it was decided to use the real-time audio and video components to develop the remote experiments.
4.2 Software Architecture

The overall software architecture for the real-time interactive remote laboratory system developed is shown in Figure 2. The system was realized using a multi-layer software approach that enables the various distributed applications (i.e. Web application, lab agent and experiment/camera controller) to interoperate with each other. Furthermore, the efficiency of the implementation and the reuse of several software components were also key concerns during the development process. The main software/hardware components are as follows:

- The first software layer is the Graphical User Interface (GUI), which presents the available information and actions to the user. The user interface is built on the assumption that users will employ all available kinds of Web browsers. Therefore, the user interface was implemented such as to be accessible from all platforms that are able to process HTTP, thus making the system entirely platform-independent.

- The second software layer is the Web application, which processes the requests from the GUI and posts back the results of these requests. Various dynamic Web content technologies can be used for these server-side applications (e.g. Common Gateway Interface (CGI)[24], Hypertext Preprocessor (PHP)[25], JavaServer Pages (JSP)[26], ASP.NET[27] and Active Server Pages (ASP)[28]). Finally, ASP.NET, which simplifies the developer’s transition from Windows application development to Web development by offering the ability to build Web pages containing controls similar to those of a Windows user interface, was selected.

- The lab agent forms the third software layer. It accepts the input from the Web application requested by the user and generates a user input queue. According to the user request, the lab agent transmits the request to the appropriate controller (i.e. experiment controller or camera controller). When the experimental procedure is finished, the lab agent records the
experiments result to the database. This layer makes the interactions between Web application, database and experiment controller transparent. The Web application does not depend on the details of the experiment implementation and the experiment controller does not depend on the details of the database operation. In our system, the lab agent locates the experiment, routes the user request, returns the experimental results and stores the results. The separation of the database and instrument operations renders the development of the Web application and the experiment controller independent, thus improving the development efficiency and making the reuse of program modules easier.

- The instrument and camera controllers constitute the fourth software layer. They are used to control the real physical instruments such as the experimental devices, lights, cameras and microphones. The control commands come from the lab agent. If a particular command necessitates some data outputs, then the controller sends these outputs to lab agent after the corresponding action has been completed. There are also some commands that do not require any data output, such as for example turning on/off the lights or activating/deactivating the camera or microphone.

- The Instruments and Cameras represent the hardware components that perform the required operations. In order to facilitate more user interaction in the system implementation, a camera with pan, tilt and zoom functions was chosen. Thus, the user can adjust the camera view based on the requirements of a particular experimental procedure. Together with the live audio/video streaming, this controllability of the camera provides the user with the feeling of physical presence at the experiment’s site.

This architecture was designed to ease and thus speed up the future development of additional remote experiments. The experiment controllers together with their associated instruments can be considered as black boxes in the system, which have a common communication interface with the system. All data (e.g. instrument status, experiment feedback, etc.) are transferred via this communication interface. Because the communication is based on platform-independent TCP/IP connections, the experiment controllers can be reused in different online learning systems. Furthermore, the adding of new experiments only requires the development of further experiment controllers, which reduces the development cost and time.

4.3 User Interaction with Experiment System

The user interaction with the system is illustrated in Figure 3, and the experiment workflow can be summarized as follows:

- Open Web connection and authenticate user
- Choose an experiment from a list of available experiments that satisfy the user requirements
- Adjust instrument settings (i.e. lighting, camera and microphone) and execute the experimental procedure (one or more times)
- Retrieve and analyze resulting data
- Close connection
Communications between the user and instructor can take place through e-mail, message boards and other tools such as Windows Live Messenger$^{20}$ and Yahoo Messenger$^{21}$. Currently, WebCT$^{29}$ is used in the system described not only for distance learning, but also as online communication platform between students and course instructors (i.e. lecturer, teaching assistants, etc.). WebCT can also be used to set up online discussion groups that allow the individual students belonging to a particular laboratory group to communicate with each other.

4.4 Message Design

For each user request, the system generates a message that passes through the various software layers (see Figure 2). The corresponding message flow is as follows (see Figure 4):

- Login: the user accesses the system
- List experiments: the Web application lists the available experiments
- Request experiment: the user selects and requests a specific experiment from the experiments list
- Adjust instrument: the user adjusts the instruments (i.e. lighting, camera, microphone, etc.) as needed
- Execute experiment: the user carries out the experimental procedure in an interactive fashion
• Return result: the instrument returns the experimental results to the lab agent
• Request result: the user request the experimental results
• Forward result: the lab agent forwards the experimental results to the user

![Message workflow diagram](image)

**Figure 4: Message workflow**

### 4.5 Experiment Execution Modes

In general, there are two scenarios for experiments, namely individual experiments and group experiments. Individual experiments are conducted by only one person. In group experiments, typically only one group member can control the experimental setup at any given time while the remaining group members may only observe the experiment. To enable both of these scenarios, a running mode and a monitoring mode of experimentation are provided by the system described here:

- To enter the running mode, the user needs to request a particular experiment and wait for the system to accept this request based on the queue for that experiment. This mode is used to carry out individual experiments and to take on the control role in group experiments. In this mode, the user has total control privilege over the experimental setup. However, to avoid any abuse of the experimental setup by occupying it for an excessively long period of time, a reasonable time limit was implemented. Once this time is expired, the user loses the control privilege over the experimental setup.

- The monitor mode mimics the observation role in a group experiment. In this mode, the user is only able to observe in a browser window the response of the experimental setup to the requests placed by the controlling user. This mode can also be employed in the context of distance education whereby the instructor can demonstrate a real experiment via the Internet while the students observe the experimental results on their computers.
5 Pilot Implementation

This new real-time remote laboratory architecture was deployed and piloted in the fall semester of 2006 in a junior level course on mechanisms and machine dynamics. This course offered at the Department of Mechanical Engineering had an enrollment of 40 students. For the first pilot of the new architecture, a mechanical vibration system previously only capable of batch-mode operation was redesigned for real-time operation. This previously existing batch-mode vibration system allows the students to explore the dynamic behavior of one-degree-of-freedom and multiple-degree-of-freedom mechanical vibration setups. Due to the unique design of the vibration devices, highly accurate displacement measurements $x(t)$ can be obtained that compare very favorably with theoretical predictions. As an example, a schematic of the one-degree-of-freedom system is depicted Figure 5. The GUI for the one-degree-of-freedom free-vibration experiment, which combines both the user information and the experimental input data into one form, is shown in Figure 6.

![Figure 5: Schematic of one-degree-of-freedom mechanical vibration system](image1)

Figure 5: Schematic of one-degree-of-freedom mechanical vibration system

![Figure 6: GUI of batch-mode remote one-degree-of-freedom free-vibration experiment](image2)

Figure 6: GUI of batch-mode remote one-degree-of-freedom free-vibration experiment
When the user selects the newly developed real-time one-degree-of-freedom free-vibration experiment, the GUI shown in Figure 7 is launched. In contrast to the GUI for the batch-mode experiment shown in Figure 6, this GUI represents only the control interface in which the experimental input (initial displacement, experiment run time, sample frequency, lighting, audio and video options) is provided, the lighting, audio and video equipment is interactively controlled and the resulting experimental output is requested, observed and stored (measured data, audio and video streams and recordings) while — contrary to the prior batch-mode implementation — the user login and user authentication are now separated. Also, in the real-time implementation, the student now has interactive control over the usage of the video and audio functions as well as the data acquisition parameters.

![GUI for free vibration of one-degree-of-freedom system](image)

**Figure 7: GUI for free vibration of one-degree-of-freedom system**

In the laboratory assignment used in the junior level course on mechanisms and machine dynamics, the students are given the values for the mass \(m\) and spring stiffness \(k\) of the system. Using these parameters, they are then asked calculate the undamped circular natural frequency \(\omega_n\), the undamped natural frequency \(f_n\) and undamped vibration period \(T\). Subsequently, they must conduct the free-vibration experiment and determine the remaining system parameters (i.e. damped vibration period \(T_d\), damped natural frequency \(f_d\), damped circular natural frequency \(\omega_d\), logarithmic decrement \(\Lambda\) and damping ratio \(\zeta\) and damping coefficient \(b\)) from the experimental data. Finally, the students have to run several additional experiments with varying initial displacement values and explore the effects that this parameter has on the system response. This
part of the laboratory assignment represents a discovery-type learning activity, which will reveal that the system response is governed by viscous damping for larger initial displacement values while Coulomb friction dominates the response for small initial displacement values.

In order to obtain proper experimental results, an appropriate input value for the sampling rate must be used, which can be selected based on the calculated undamped vibration period $T_n$. Figure 8 shows two sample displacement-vs.-time plots for the same initial displacement but different sample rates, whereby it is clearly seen that an inappropriate choice for the sampling rate causes misleading results (see left plot). Alternatively, a trial-and-error approach can be employed to adjust the sampling rate until reasonable data for the displacement as function of time are obtained.

![Displacement versus time](image)

Figure 8: Samples of the displacement vs. time plots

6. Conclusions

The rapid spreading of broadband Internet access is enabling new methods of delivery for modern engineering and science curricula. Remote and virtual laboratories are beginning to play an increasing role in education and training. This paper describes the design and implementation of a remote laboratory architecture, which allows the execution of experiments in real time. Contrary to previous remote-laboratory implementations based on a batch mode of operation, in this new system the students can interactively control the experiments and obtain the corresponding outputs (including raw data, data plots, video/audio streams and recordings) in an integrated browser-based GUI. Furthermore, this real-time system was designed in a platform-independent fashion such that it will facilitate its expansion beyond the boundaries of the original institution. As an example for the implementation of such real-time interactive laboratories, the redesign of a preexisting batch-mode remote experiment using a vibration system is described. This system was successfully piloted in a junior-level course on mechanisms and machine dynamics in the fall of 2006. While the general student feedback after this pilot implementation was very positive, a more detailed assessment of the learning effectiveness of remote experiments is planned for the Spring 2007 semester.
Acknowledgments

This work was supported by NSF Grant No. 0326309. This support is gratefully acknowledged. The collaborative efforts and discussions with Mr. Chenghung Chang are very much appreciated.

References

[18] RealNetworks at: http://www.realnetworks.com/
[23] Lakeofsoft at: http://www.lakeofsoft.com/
[24] Common Gateway Interface (CGI) at: http://www.w3.org/CGI/
[27] ASP.NET at: http://asp.net/
[29] WebCT by Blackboard Inc. at: http://www.webct.com/