A WEB-BASED OPERATED EXPERIMENT FOR UNDERGRADUATE ENGINEERING LABORATORY

Jing Guo, David J. Kettler, Muthanna Al-Dahhan*

Energy, Environmental and Chemical Engineering (EECE) Department
Washington University, St. Louis, MO, 63130

ABSTRACT

An undergraduate experiment in the Department of Energy, Environmental and Chemical Engineering, Washington University in St. Louis, has been developed for on-line operation over the Internet. By adopting rapidly changing Internet and object component technologies, we developed a novel approach combining the Internet and regular laboratory equipment. The client–server applications use a Visual Basic and Labtech programming environment, which includes remote access of control and data acquisition. The details of the architecture design and the experiment are discussed to provide enhanced understanding of the technical issues involved and the solutions chosen.

***Index Terms*:** **Distance Learning, Internetworking, Remote Operation, On-line Laboratory, Server/Client Application, Tracer Response**

The corresponding author is with the Department of Energy, Environmental and Chemical Engineering, Washington University, Saint Louis, MO, 63130, USA (Phone: 314-935-7187; fax: 314-935-7211; e-mail: muthanna@wustl.edu).

INTRODUCTION

Practical experimentation that processes real signals is essential to help students understand theory given in textbooks and give them skills to deal with real problems successfully. An indispensable part of the engineering curriculum is the laboratory class, which is designed to acquaint the students with practical experience and train them at the same time in an effective way for acquiring face-to-face interaction. This conventional approach, however, imposes difficulties on students with time or distance constraints. Moreover, due to both safety and security reasons, the access to the labs cannot be totally free and is restricted in time in order to ensure the presence of supervision personnel. Interesting approaches have been suggested to use the Internet for various educational purposes, including different types of virtual laboratory Web sites [1], interactive simulations [2], and access to real instruments and test benches through a remote connection [3-5]. In fact, some implementations of remote monitoring and control through the Internet have already reached the teaching laboratories of physics [6] and electrical engineering [7]. For chemical engineering laboratories, this capability is currently available at University of Tennessee at Chattanooga [8], University of Texas at Austin [9], MIT [10] and Washington University in St. Louis (the experiment discussed here).

With appropriate planning, teachers and students can run the Web-connected experiments on a flexible schedule, which provides educational facilities and opportunities for those students whose schedules might be asynchronous [8,11]. Another advantage of such remotely accessible laboratories is that teachers and students at another institution can have access to laboratory facilities without incurring the full cost of developing such resources. Rather than several universities spending money on the same equipment for the same experiments, cooperating universities may each carry out one unique experiment and then form a pool of experiments [12,13]. Using such highly automated experiments for remote operations can allow a drastic reduction in the amount of personnel time required for those particular experiments. It is reported that online laboratories hold promise of being up to two orders of magnitude cheaper than conventional ones [14]. Having access to tutorials, pictures, videos, past data files, data processing tools, and graphs tracking the dynamic process variables, these Web pages provide students with sufficient resources that can be viewed simultaneously by all laboratory class members [10]. Such expanded access allows the students and instructors spending less time communicating the operating procedure and more time investigating the experimental results. Remote learning has evolved into a new model with high quality aimed at engaging students in a distinctive learning technology that helps build a solid foundation [9, 15].

Advances in the computer software and interfacing techniques enable remotely operated laboratory experiments to be constructed at relatively low cost [16]. In this work, an in-house automated and internet based operated experiment has been developed for characterizing a tubular reactor via tracer measurement method. This experiment is offered to the undergraduate students in the Department of Energy, Environmental and Chemical Engineering at Washington University in St. Louis. A client-server architecture devoted to the instrument management through the Internet is built with Visual Basic and Labtech programming tools, providing a novel approach in comparison to the Java and Labview software employed in other references [8-10]. The architecture is described along with the specification and design of a geographically distributed system based on standard commercial components. Used for the required undergraduate chemical engineering unit operation laboratory course, a tracer experiment is restructured to illustrate the connection between physical instruments and the server-client Internet system. The experimental data are archived for subsequent viewing and analysis, and the responses of students to the developed Internet based experiment are assessed.

SYSTEM ARCHITECTURE

To achieve a standard component distribution system, we adopted WWW technologies, which allow portability and independence through different client hardware/software architectures. A standard portable language is instrumental for independence of the application from the client system on which it is executed. An Internet browser can now be considered a standard component of any computer installation. Therefore, our approach will automatically work with any widely available hardware/software environment.

The connection between the server and client program is made by a Transmission Control Protocol/Internet Protocol (TCP/IP) Winsock socket located within both programs, which functions much like a phone receiver/dialer on each side of the Internet. The server and the clients are connected on the same local area network (LAN) within the laboratory or campus. Remote connections can even be set up between the server and a single user working at home.

TCP/IP defines the physical interconnection, data transfer, and message routing management. It is the typical protocol suite adopted in the standard Internet [1]. The server sends measurement data to the client the same way the client sends control commands to the server, by creating a string of numbers representing all the commands or measurements and sending them through the TCP/IP socket. Likewise, once the client receives the measurement string of numbers and the server receives the control string of numbers, the string is parsed, and each measurement or command within the string is sent to its appropriate subroutine within the client or server code.

A block diagram of the proposed solution is shown in Figure 1. The clients are hosted on a user’s personal computer while the server runs on a laboratory computer and manages an automatic control and measurement system that embeds programmable instruments. Both client and server computers run programs which are logically split into two layers. One layer in both client and server sides deals with user interface and instrument management, while the other layer deals with network intercommunication. The server is directly connected to the instruments that measure physical quantities. In this work, the server computer, connected physically to the instruments, makes available a set of remotely callable procedures that perform all standard activities (address, read, write, status polling, etc.). The client's command generator issues commands according to the parameter set specified by the user, and sends them via the TCP/IP client socket to the server. The experimental results sent back by the server are then handled and displayed in the client window. The sockets of the client program and the server program are connected by using the server computer's IP address on the Washington University network. The same local port number must be specified within both the client and server sockets. Socket connections and the TCP/IP communication protocols transmit the instrument control commands, parameters, and reports between client and server.

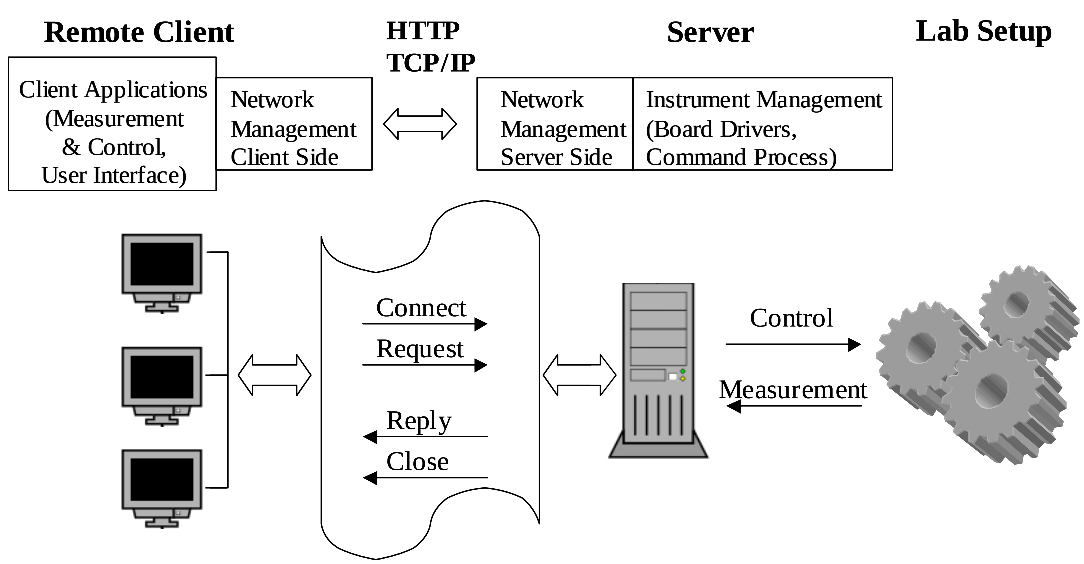

Figure 1. Diagram of the client-server architecture employed to implement the remote control-measurement

Only one user group is allowed to connect at one time, because physically only one experimental run can be performed on the experimental setup at a time. At the session's termination, the socket connection is closed and the server can accept a new connection on the same port to start a new session. If the client/server connection is broken or remains idle over 5 minutes, the server application shuts down the system. If the power shuts down, a system of safety interlocks in the physical system prevents the system from running indefinitely. To protect the server, several techniques can be used, e.g., access restriction based on user identification, firewalls, and encryption. For simplicity and cost reasons, we adopted an approach based on access restriction through user verification of both the password and the IP address of the gateway through which client connects to the server.

On the server site, the structure offers great flexibility. Developed in Visual Basic, the programs require only the addition of a very small number of statements necessary for establishing and closing the interface-related functions of corresponding network functions. When new instruments are added to the instrument library, it is easy to add a measurement or control variable with small modifications to existing programs. The software related to any newly connected equipment can be added to the system without recompiling or modifying the application core. About two or three lines added to the server and client programs will add numbers representing the additional measurements or control variables to the string sent through the TCP/IP socket.

At the client site, because the whole core of the software application (i.e., the components required to share, engage and release the resources) resides permanently on the server computer, it is not necessary to install any special software tool. Once a new client connection is accepted, the user locally runs the command necessary for selecting and driving an instrument. As a consequence, the proposed structure makes the application portable and safe for remote users.

The server is a Pentium-IV computer with a 1 GHz processor, provided with two independent Universal Serial Bus (USB) ports. The server runs on Windows 2000 Professional and uses drivers from Data Translation to access the Data Translation DT9804 interface board. The server connects to the interface board using a USB cable, and the interface board has analog input/output and digital input/output ports for connection to the physical control hardware of the reactor system. The overall system has been devised to assure reliable communication between the client and the server, and between the server and the physical resources available. A Labtech ControlPro 12.1 Runtime program receives data and sends commands to the control hardware via the DT9804 interface board. Once the control hardware and server are physically connected to the DT9804 board and the board's drivers are installed, Labtech can be set up to control the hardware by dragging-and-dropping control icon blocks from its menu into its workspace.

BUILDING EXPERIMENTAL INSTRUMENTS

The lab setup icon shown in Figure 1 represents any real instrument that requests automatic control and measurement. The proposed server-client system structure can be extended to fit a variety of requirements and serve different experiments. As a test case performed on an implementation of the whole system, a tracer study experiment is carried out remotely in real-time over the Internet, to assess the flow pattern of a tubular reactor as one of the experiments for the Chemical Engineering Laboratory class at Washington University in St. Louis. The purpose of the tracer test is to experimentally determine the deviation, if any, from the ideality of a real continuous flow reactor.

In an ideal plug flow reactor, the fluid flows through the reactor with a piston-like motion and no axial mixing. A real tubular reactor, however, cannot reach this ideal state. In the experiment, a conductive tracer was injected into water just before the reactor entrance, and the conductivity of the solution mixture was measured at the reactor entrance and exit. The mean residence time, tracer response curve variance, dimensionless variance, and axial dispersion coefficient can be calculated from the measured tracer response curve. Through data analysis, the tubular reactor was compared with an ideal plug flow reactor.

Figure 2 shows the schematic diagram of the physical hardware built for the tracer study experiment. The students open the feed flow valve, adjust the water flow rate by varying valve position, start the tracer feed pump, adjust the tracer injection duration, open the tracer injection valve, and then inject the tracer. Air pressure (30-100 psi) is used as the driving force to control the feed flow valve and the tracer injection valve. The feed flow rate is measured by a turbine flow meter downstream, and the measurement is sent to the client application. The conductivity measurement will rise and then fall back to the steady state value, at which point the students may close the client application.

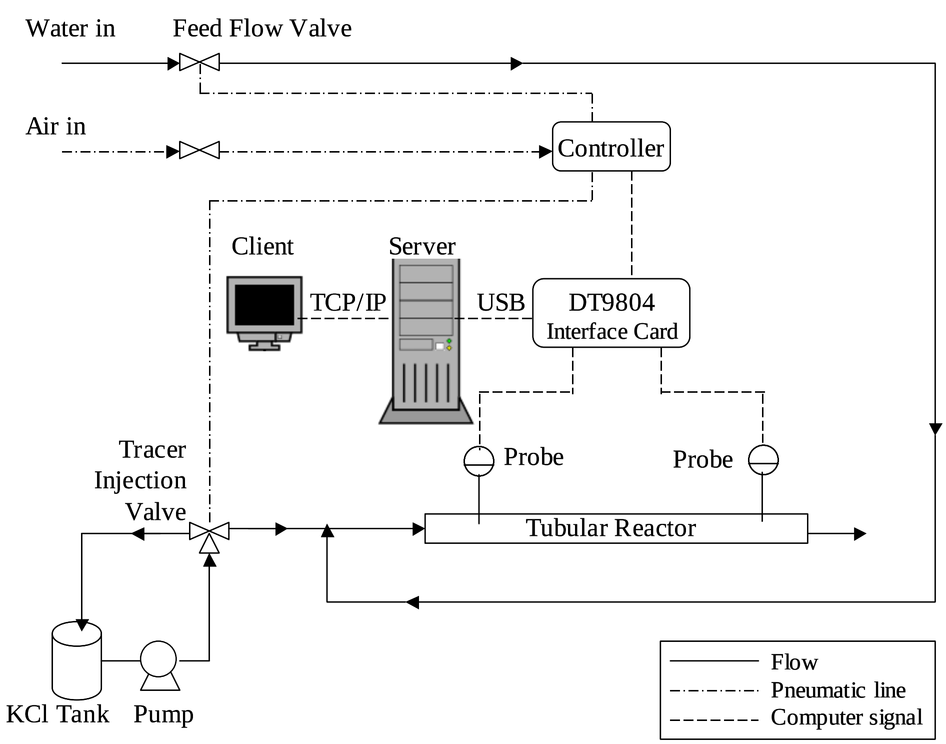

Figure 2. Overview of the physical experimental setup and its connection to the Web

Table 1 lists all of the process variables used as signals in the tracer experiment. Digital signals are either on or off when equal to 1 or 0, respectively. Analog signals send (Output) or receive (Input) signals within a defined range of values, shown in the "Range" column in Table 1. The output signals control the instrument setup, while the input signals are variable measurements obtained at a specific condition. The types of all variable signals are listed in the "Type" column. The elements that launch and receive signals are listed in "Origin" and "Destination" columns, respectively.

Table 1 Process Variables Used in the Tracer Experiment

Variable	Range	Units	Type		Origin	Destination
Reactor Feed Valve	0-1	Volts	Digital	Output	"Client_TracerStudy.exe"	Control Hardware
Tracer Feed Pump	0-1	Volts	Digital	Output	"Client_TracerStudy.exe"	Control Hardware
Inject Tracer	0-1	Volts	Digital	Output	"Client_TracerStudy.exe"	Control Hardware
Tracer Injection Duration	1-3	sec	Analog	Output	"Client_TracerStudy.exe"	"Server_TracerStudy.exe"
Reactor Feed Valve Position	0-100	%	Analog	Output	"Client_TracerStudy.exe"	Control Hardware
Run Time	>0	sec	Analog	Input	"Traceexe.ltc"	"Client_TracerStudy.exe"
(Reactor) Influent Conductivity	>0	mS	Analog	Input	Control Hardware	"Client_TracerStudy.exe"
(Reactor) Effluent Conductivity	>0	mS	Analog	Input	Control Hardware	"Client_TracerStudy.exe"
(Reactor) Feed Flow Rate	>0	L/min	Analog	Input	Control Hardware	"Client_TracerStudy.exe"

NETWORKING CREATION FOR ON-LINE CONTROL AND MEASUREMENT

The first application in the server computer is the Labtech Runtime program "Traceexe.ltc". It initiates the connection between the physical laboratory setup and the server computer. This connection channel receives measurement signals from the USB port on the interface card and issues commands to control the setup operation. All measurement variables are classified as analog inputs in “Traceexe.ltc” program. By specifying the correct interface point, each analog input block in “Traceexe.ltc” receives the proper signal from the interface card.

The second application in the server computer, “Server_TracerStudy.exe”, activates the server site and enables it not only to transfer the remote client signal to the physical setup, but also to receive measurements from “Traceexe.ltc” by continuously using the GetLT function. This function uses a built-in Labtech application called LT-Speedway to “grab” the analog input data received by “Traceexe.ltc”. The program “Server_TracerStudy.exe” takes four measurement variables it receives from its GetLT function and combines them into one string of text, called “OutputString”. Once the client connects to the server computer using the client program “Client_TracerStudy.exe”, “Server_TracerStudy.exe” sends “OutputString” across the Internet once every 100 milliseconds to the client program, using a timer within the server program called “Timer2”.

These two applications must be running before a student can access the experiment using the client program, “Client_TracerStudy.exe”. The student downloads this program from an Internet page and stores it on the remote computer. Once the student double-clicks on the related icon, the client program opens up and connects to “Server_TracerStudy.exe” on the server. Every command the user manipulates sends text data from the client TCP/IP socket across the Internet to the server TCP/IP socket. The server program sends measurement data acquired from the Labtech Runtime program “Traceexe.ltc” back to the client through TCP/IP socket and sends the control variable commands acquired from the client to “Traceexe.ltc”, where it is executed by that program on the control hardware.

The controlled variables, reactor feed “Valve Position” and tracer “Injection Duration”, are analog outputs in “Traceexe.ltc”. Before injecting the tracer, the student sets the values of analog outputs by using a scroll bar on the user interface of the client program, “Client_TracerStudy.exe” as shown in Figure 3. The student must also open the reactor “Feed Flow Valve”, turn on the “Tracer Feed Pump”, and “Inject Tracer” by pushing the respective buttons on the client user interface. These are digital outputs in “Traceexe.ltc”. Whenever the student pushes one of the buttons (digital) or slides one of the scroll bars (analog) on the client user interface, the current values of all the digital outputs are combined with the analog outputs as a text string called “InputString” in the InputData function in the client program. Then “InputString” is sent to the server program through the client TCP/IP socket. The server program picks up the “InputString” of text across the Internet at its TCP/IP socket, separates all of the outputs, and places them in their respective textboxes, as shown on the left hand side of the image in Figure 4. In the server application, the function PutLT takes each of the values in the textboxes on the left hand side of the server monitoring window and sends them to their respective input block in “Traceexe.ltc”. The input blocks receive each signal in “Traceexe.ltc” and send them to their corresponding Bit Number (digital) or Interface Point (analog) on the DT9804 interface card.

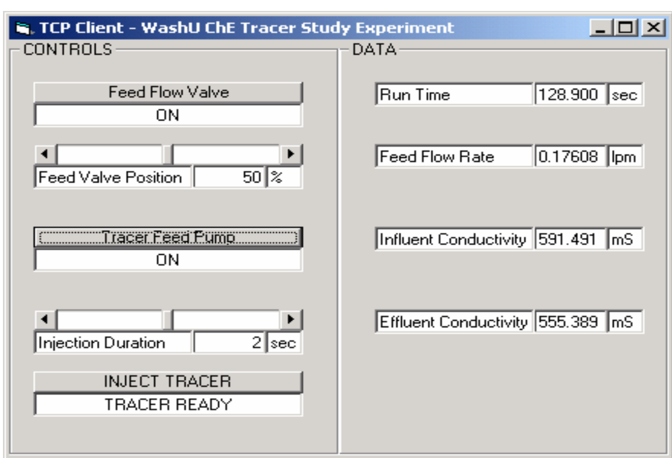

Figure 3. User interface for the client application

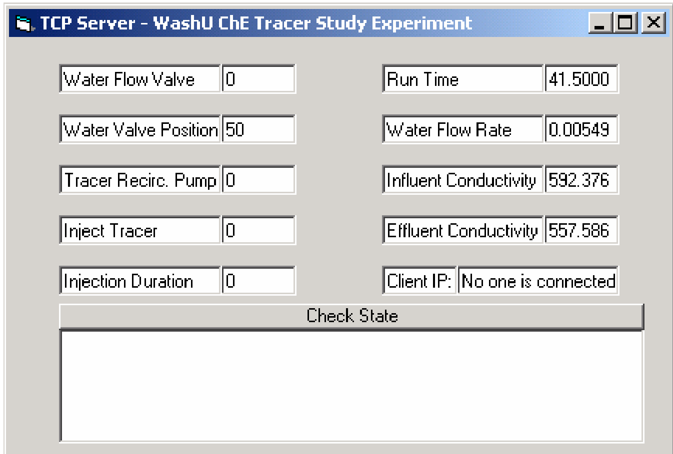

Figure 4. Monitoring window for the server application

LABORATORY EXPERIENCE

The experiment website is "http://it.che.wustl.edu/che473". After a class of introduction to the distributed learning technology and two additional classes on the theoretical aspects of the experiment, the instructor demonstrated and monitored the experiment using a classroom computer connected to the Internet. When the students were executing the measurements themselves in the computer lab one floor above the undergraduate unit operation laboratory, they were asked to provide their inputs on the user-interface, analyze the experimental outcomes, and answer questions posed by the instructor through the interactive dialog. Explanatory web pages were provided to answer most of their questions on the real instruments arising during the lab session. As a result of this interactive tutoring mode, students showed more interests in the on-line operation than the local on-site operation.

During the lab session, students issued commands and parameters from the client window to the server via the TCP/IP client socket. The experimental results were sent back by the server and then handled and displayed in the client window. The client program created a log containing measurements of time, flow rates, influent conductivity, and effluent conductivity.

Once the client application is closed, students can open this log to analyze the evolution of the collected tracer response. Typical tracer response curves at the inlet and outlet with respect to time are shown in Figure 5. In an ideal plug flow reactor, the tracer curves collected at the exit and entrance would be identical as thin, spike like peaks. This experiment, however, found that dispersion and stagnancy have significant effects on the tracer response curves. Therefore, reactor's nonideality is necessary to be included in the reactor scale model in order to properly predict reactant conversion from given feed rates and reactant compositions.

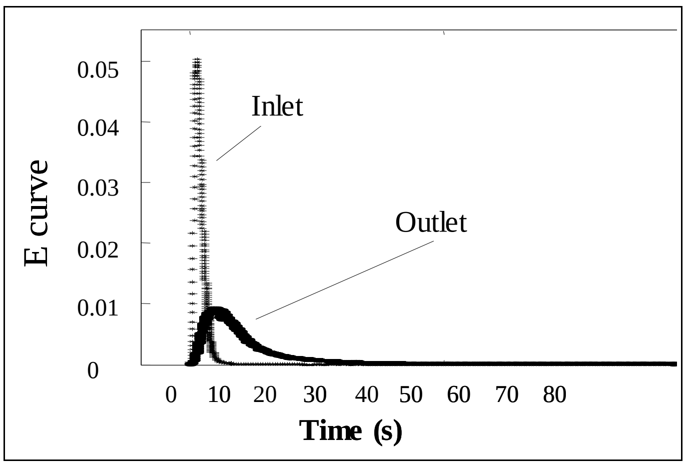

Figure 5. Typical tracer response curves shown in the client side. Measurements are taken at the inlet and outlet of the tubular reactor

The experiment was conducted twice, firstly at a remote client computer station that was Internet-linked to the server (remote control), and secondly at the server computer station directly attached to the setup elements (local control). The typical experimental values obtained from the online remote control and on-site local control, as well as the relative error between them, are listed in Table 2. Although there is a time delay between the client and server due to the web data transfer and the instrumentation synchronization, this delay penalty is negligible when instruments take a long time to complete the measurement [4]. Hence, the insignificant relative error leads us to conclude that the online control and on-site control give rise to same residence time measurement in the tubular reactor.

Table 2 Experimental Measurements of Residence Times

Valve opening position (%)	Actual Flow rate (cm/sec)	Local Control (sec)	Remote Control (sec)	Relative Error (%)
45	2.93	22.96	23.77	-3.53
55	3.86	18.32	19.02	-3.82
65	5.11	14.57	14.97	-2.75
75	6.02	11.90	12.19	-2.44
85	6.44	11.74	11.38	3.07
95	6.67	10.87	10.67	1.84

Student feedback is a key consideration for improvement of the experiment. Surveys were filled out by the students after each lab session to evaluate the beneficial features of remote learning experience and the fulfillment of educational objectives. The responses contained encouraging comments and constructive suggestions. In general, the proposed survey recommendations were implemented before the next student group was invited to perform and evaluate the developed experiment. Students agreed that the lab sessions should be improved with more user-friendly options and tools added to the client window. One feature the students liked most about operating experiments remotely was that it allowed them to perform the process at any time from a place that was convenient for them. The other appreciated feature was that the remote operation helped the students get used to a real world application that was either in a remote control room or at a remote operation facility, especially when hazards and safety concerns were present. Some students showed desire to run the experiment on-site as a complementary reference check since their understanding of the experimental flow scheme can be enhanced with the actual devices in front of them. Actually this desire could be fulfilled by incorporating the live audio and video streaming to the remote client window so that the students can listen to the sounds of the device station and view it from different angles on the Internet while they are operating. Such sophisticated user interface can be added to the current system.

CONCLUSION

An Internet-based, client-server architecture specifically designed to allow flexible management of remote instruments has been developed. The proposed solution is portable via the employment of the TCP/IP protocol suite. It is also extensible because of the high level of abstraction in system implementation. This approach offers a valuable component to remote engineering instruction that cannot be replaced by simulation software packages. Compared to the traditional way of teaching, due to the absence of schedule and physical constraints, this new approach reaches students who otherwise would not have chance to take these courses and allows a larger and more diversified audience to access learning opportunities. The proposed technique for the control of remote instrumentation can be developed for other experiments related to the emerging field and technologies such as renewable and sustainable energy, chemicals and materials production, environmental processes, pharmaceuticals and bio-processing. In addition, there is the opportunity to use this technology to add experimental demonstrations or assignments to other courses. In order to expand the scope of the experiments and to share costs and software development time, we are offering collaboration with other universities on such approaches.

Acknowledgement

The authors acknowledge the technical help provided by Mr. Steve Picker. We also thank Professor M. P. Dudukovic (department chair at the time of this development) for his support and encouragement.

Biographical Sketch

Jing Guo received his Doctoral degree in Chemical Engineering from Washington University in St. Louis in 2004, where he worked on the experimentation of catalysis in multiphase reactors, including trickle bed reactor and packed bubble column. He also developed models and related computer programs to simulate the multiphase reactions for the applications ranging from bench to commercial scales. Jing Guo received his Bachelor degree in Chemical Engineering in 1997 and his Master degree in Chemical Engineering from Beijing University of Chemical Technology in 2000. He is currently with UOP, Chicago.

David J. Kettler received bachelor degrees in Biomedical and Chemical Engineering from Washington University in St. Louis in 2001. During his study, he was also responsible for developing the Process Control Laboratory’s homepage and the Simulink Virtual Laboratory as an interactive series of workshops.

Muthanna H. Al-Dahhan is Professor of Energy, Environmental and Chemical Engineering at Washington University in St. Louis and Co-director of the Chemical Reaction Engineering Laboratory (CREL). He received his Bachelor degree in 1979 from University of Baghdad in Iraq, Master degree in 1988 from Oregon State University and Doctoral degree from Washington University in 1993; all in Chemical Engineering. His research interests are in the fields of multiphase reaction engineering with development and implementation of advanced measurement techniques and modeling of transport-kinetic interaction for design and scale-up of multiphase reactors and processes for production of clean and alternative fuels and chemicals, bioprocesses, biomass conversion, energy/bio-energy, and waste treatment. He is an author of more than 100 peer-reviewed papers.

References

[1] A. Ferrero, and V. Piuri, "A simulation tool for virtual laboratory experiments in a WWW environment," *IEEE Trans. Instrum. Meas.*, vol. 48, pp. 741-746, 1999.

[2] D. Shin, E. S. Yoon, S. J. Park, and E. S. Lee, "A web-based, interactive virtual laboratory system for unit operations and process systems engineering education: issues, design and implementation," *Computers and Chem. Eng.*, vol. 26, pp. 319-330, 2002.

[3] L. Benetazzo, M. Bertocco, F. Ferraris, A. Ferrero, C. Offelli, M. Parvis, and V. Piuri, "A web-based distributed virtual educational laboratory," *IEEE Trans. Instrum. Meas.*, vol. 49, pp. 349-356, 2000.

[4] M. Bertocco, F. Ferraris, C. Offelli, and M. Parvis, "A client–server architecture for distributed measurement systems," *IEEE Trans. Instrum. Meas.*, vol. 47, pp. 1143-1148, 1998.

[5] P. Arpaia, A. Baccigalupi, F. Cennamo, and P. Daponte, "A measurement laboratory on geographic network for remote test experiments," *IEEE Trans. Instrum. Meas.*, vol. 49, pp. 992-997, 2000.

[6] C. L. Enloe, W. A. Pakula, G. A. Finney, and R. K. Haaland, "Teleoperation in the undergraduate physics laboratory-teaching an old dog new tricks," *IEEE Trans. Educ.*, vol. 42, pp. 174-179, 1999.

[7] H. Shen, Z. Xu, B. Dalager, V. Kristiansen, O. Strom, M. S. Shur, T. A. Fjeldly, J. Q. Lu, and T. Ytterdal, "Conducting laboratory experiments over the Internet," *IEEE Trans. Educ.*, vol. 42, pp. 180-185, 1999.

[8] J. Henry, "Web-Based laboratories: technical and pedagogical considerations," *AIChE Annual Meeting*, Reno, NV, 2001.

[9] L. Rueda, T. F. Edgar, "Process dynamics and control experiments carried out over the Internet," *AIChE Annual Meeting*, San Francisco, CA, 2003.

[10] C. K. Colton, "Heat exchanger experiment at MIT," *International Conference on Engineering Education*, Valencia, Spain, 2003.

[11] H. A. Latchman, C. Salzmann, D. Gillet, and H. Bouzekri, "Information technology enhanced learning in distance and conventional eduction," *IEEE Trans. Educ.*, vol. 42, pp. 247-254, 1999.

[12] I. T. Cameron, "An interactive web-based decision support system for hazardous industry land-use planning," *Computers and Chemical Engineering*, vol. 24, pp. 1057-1062, 2000.

[13] D. Shin, E. S. Yoon, S. J. Park, and E. S. Lee, “Web-based interactive virtual laboratory system for unit operations and process systems engineering education,” *Computers and Chemical Engineering*, vol. 24, pp. 1381-1385, 2000.

[14] W. Aung, P. Hicks, L. Scavarda, V. Roubicek, and C. H. Wei, “*Engineering education and research: A chronicle of worldwide innovations*,” Arlington, VA, USA: iEER., 2001.

[15] M. Hough, and T. Marlin, “Web-based interactive learning modules for process control,” *Computers and Chemical Engineering*, vol. 24, pp. 1485-1490, 2000.

[16] A. Ferrero, S. Salicone, C. Bonora, and M. Parmigiani, “ReMLab: A Java-based remote, didactic measurement laboratory,” *IEEE Trans. Instrum. Meas*., vol. 52, pp. 710-715, 2003.