Remote Access Laboratories for Engineering Education

Authors:

M. Reza Emami* and Michael G. Helander
Institute for Aerospace Studies, University of Toronto,
4925 Dufferin Street, Toronto, Ontario, Canada, M3H 5T6

Abstract — *This paper discusses the development and implementation of remote access laboratories for engineering undergraduate programs. It begins by addressing the needs for and benefits of remote access laboratories and reviewing current developments at different universities. It then presents a generic and scalable architecture for a portal that hosts the student accounts and connects them to the target workstations running the experimental apparatuses and to a number of software applications that students need for analyzing the experimental data. Next, it details the technologies used to implement this architecture including the software and hardware components. Finally, the paper concludes by reporting the students' usage of these experiments and analyzing their feedback, and discusses whether and how remote access laboratories can bring benefits to the engineering pedagogy and ways of minimizing their shortcomings and potential pitfalls.*

Index Terms — *Distance learning, eCollaboration, eLaboratory, eLearning, Engineering portal, Web-based education.*

1. INTRODUCTION

Engineering pedagogy heavily relies on laboratory experimentation. Students, particularly at the undergraduate level, develop engineering intuition by continuously iterating between mental concepts and physical hardware. The experiential learning process, however, is iterative in that observations are made, analyzed and then new data are collected based on the results of the analysis until a clear understanding is gained. Consequently, instructional laboratories must ideally provide open and flexible schedules for the students to perform experiments in various durations and multiple times. This requirement is currently beyond the capacity of most schools. Students' access to the laboratory equipment and experimental setups has traditionally been limited to definite sessions at particular hours and locations, due to the shortage of time and resources imposed by the continuously rising engineering enrolment. As a result, struggling students who need more time to comprehend the relevance and significance of the laboratory exercises are constantly frustrated by such constraints throughout their education. Furthermore, many engineering schools around the globe, despite providing rigorous theoretical courses, cannot afford adequate modern experimental laboratories that expose their students to state-of-art instrumentation supported by computerized data acquisition, control, analysis, and presentation. Therefore, with the advent of computer and Internet technologies it has become appealing to both researchers and pedagogs to study the methods and implications of remotely-accessible instructional laboratories, in the hope of utilizing expensive specialized equipment more effectively and providing scheduling flexibility regardless of time and geographical location.

In addition, the merit of remote access laboratories is beyond a merely new mode of experimentation. The newly-revised engineering curricula have begun to recognize the need for the diversity of scope, expertise, and even resources in engineering education. A multifaceted curriculum aims at training engineers who can work at multinational corporations in teams composed of members with a wide range of expertise, technical and cultural background. Therefore, the formation of inter-disciplinary, inter-university engineering programs has changed course from wishful thinking to serious planning. Consequently, most engineering programs have begun to integrate web-based education, usually labeled as *eLearning*, into the traditional curricula both heuristically [1] and systematically [2] to cover a wider population, and remote access laboratories have started to serve as platforms for inter-university collaborations aimed at establishing global distance education consortia [3].

Remote access laboratories have been extensively addressed in the literature from both the viewpoints of their technical feasibility and their implications for engineering pedagogy. A significant number of remotely-accesible experiments have been deployed globally across all the major engineering disciplines through telepresence [4], teleoperation [5], and telecontrol [6]. Notably, Ferrero *et al.* [7] implemented a remote electrical engineering laboratory using Java™ applets to control a remote data acquisition system used to measure and characterize a number of common electrical components. Chang *et al.* [8] developed a remotely-accessible photonics laboratory using custom ActiveX® controls, written in Visual

* Email: emami@utias.utoronto.ca

Basic®, to control a photonics test bed and to receive streaming video from a number of webcams. Tang *et al.* [9] implemented a remote access laboratory for Intrusion Detection Systems (IDS) training using the Microsoft® Remote Desktop Web Connection client to provide browser-based access to remote network test stations. Wirgau *et al.* [10] developed a remote access laboratory for civil engineering using a real-time Data Acquisition (DAQ) board to control a shake table experiment using a LabVIEW™ interface.

A few groups have reported on the success of deploying remotely-accessible experiments alongside a Learning Management System (LMS). Wirz *et al.* [11] developed a distributed remote multi-robot laboratory with learning support provided by the *Moodle* [12] courseware package. A more comprehensive architecture was introduced by Rapuano *et al.* [13], using a network of servers and Target Personal Computers (PCs) to host a remotely-accessible measurement laboratory as part of an *eLearning* system. Additional examples of remote access laboratories include [14-19].

Several papers have also reported on preliminary studies into the effectiveness of remote access laboratories in comparison to traditional proximal laboratory instruction. Corter *et al.* [20] evaluated proximal versus remote experimentation using student feedback surveys, and composite test scores. Lindsay *et al.* [21] conducted an in-depth systematic study of the learning outcomes for different laboratory access modes.

Despite the increasing interest in the field, almost all remote access laboratories previsously reported in the literature lack the social constructs and collaborative interactions that transcend any particular classroom, laboratory or course. To overcome this limitation a new paradigm is needed that is an *extension* of the traditional instructional laboratory to a web-based domain. This paradigm can be realized through the concept of a corporate intranet portal. In general, a portal is a web system that provides an intuitive and personalized gateway to resources on a network. In the past, portals were synonymous with search engines that provided links to websites on the internet. More recently, however, portals have expanded to include both the aggregation of resources and the platform for building a collaborative user environment. Since this new paradigm is part of the much larger *eLearning for Engineering* introduced by the authors in [22, 23], this new class of remote access laboratory is labelled as *eLaboratory*.

The paper first presents a generic and scalable architecture for a portal that hosts the remote access laboratory resources and provides a constructivist medium for collaborative learning in Section 2. Section 3 details the technologies used to implement this architecture including the software and hardware components. Section 4 describes the evaluation methods and tools used to quantify the performance of the implemented architecture and the results of the students' evaluation. Finally, some concluding remarks are made in Section 5 with highlights of the future research.

2. eLaboratory Architecture

The eLaboratory architecture, a new paradigm for remotely-accessible instructional laboratories, is functionalized into five integrated modules; namely the Remote Experiment module, Engineering Portal module, Telepresence module, Application Publishing module, and Scheduling module, as depicted in Figure 1. The generic modular design makes the eLaboratory architecture applicable to all engineering disciplines, and allows new installations to be rapidly deployed and reconfigured.

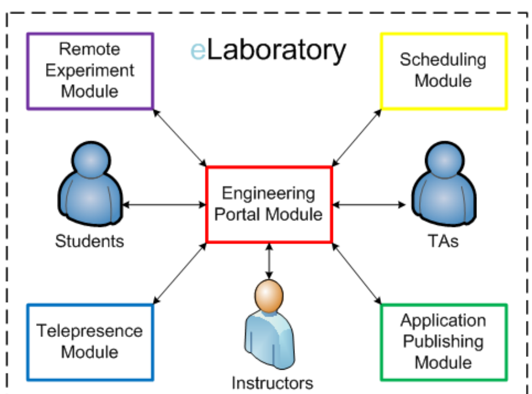

FIGURE 1

SCHEMATIC OVERVIEW OF THE ELABORATORY ARCHITECTURE.

∗ Email: emami@utias.utoronto.ca

2.1 Remote Experiment Module

The remote experiment module, illustrated in Figure 2, is used to provide students with access to hardware resources, and is the primary means of delivering experiential education.

The physical hardware for each experiment consists of an experiment setup that has been modified for computer-aided data acquisition and control using electromechanical actuators and numerous analog and digital transducers. The hardware interface layer connects the transducers and actuators from the experiment setup to the logical control software. The heart of the hardware interface layer is a workstation, called the Target PC, which is equipped with one or more DAQ boards.

The software interface layer connects the hardware layer to the user environment. It is comprised of three components; the Operating System (OS), Graphical User Interface (GUI) and Watchdog. The OS running on the Target PC supports the other components of the software interface layer and provides a central point from which to manage each experiment. The GUI for each experiment provides an interactive user friendly environment for controlling the physical experiment hardware. Figure 3 depicts the user interface for the subsonic airfoil experiment. The watchdog monitors the entire system to protect the physical hardware from incorrect user input or software failures.

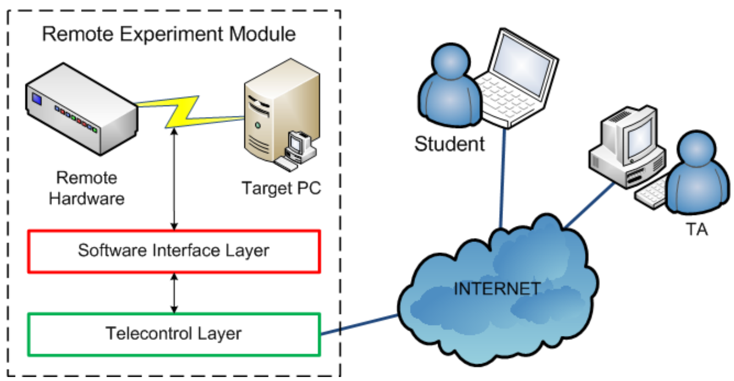

FIGURE 2

SCHEMATIC OVERVIEW OF THE REMOTE EXPEXPERIMENT MODULE.

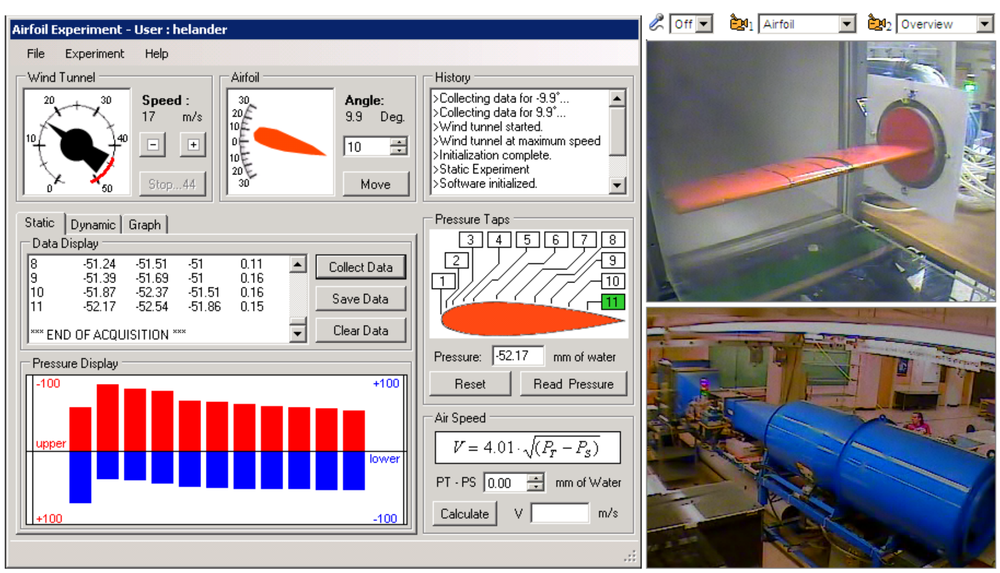

FIGURE 3

USER INTERFACE FOR THE SUBSONIC AIRFOIL EXPERIMENT.

∗ Email: emami@utias.utoronto.ca

The telecontrol layer hosts remote access to the software interface on the Target PC via the Remote Desktop Protocol (RDP). RDP is a thin-client protocol for channeling a desktop session from a remote server to a local machine. An RDP thin-client transmits mouse movements and keystrokes to the remote server, while simultaneously displaying screen data from the desktop session.

2.2 Engineering Portal Module

The engineering portal module, illustrated in Figure 4, is analogous to the physical space of a traditional instructional laboratory. It is the central web-based gateway that enables students, instructors and Teaching Assistants (TAs) to access the eLaboratory.

The web interface layer is a collection of dynamic templates that dispaly various content sources using reusable wrappers called Web Parts. Web Parts allow novice users to build a web site from a collection of distributed content and resource without having to code even a single line of Hypertest Mark-up Language (HTML).

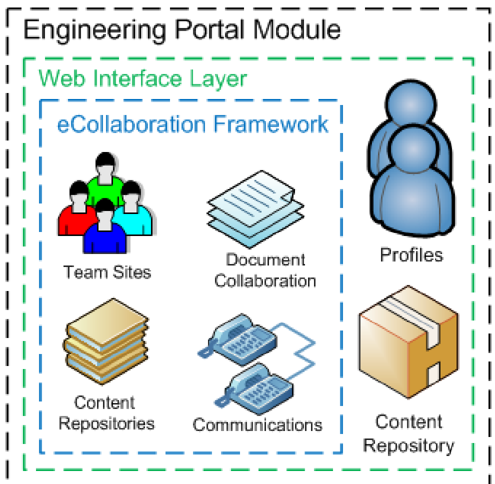

FIGURE 4

SCHEMATIC OVERVIEW OF THE ENGINEERING PORTAL MODULE.

The eCollaboration framework is the architectural core of the engineering portal. It provides students with contextual-based learning tools through a combination of communication and resource-sharing components. Each student is given a personal site on the portal—analogous to a lab notebook—for storing and managing their documents and laboratory work. Students can select portions of their personal site to publish to the portal, or post on the internet.

The portal contains a directory of all registered users that is searchable based on user profile entries, similar to a phone book. A group of students can use the directory to find each other and self-organize into a team for completing a group assignment. These groups can then create team sites in the portal on a self-serve basis using dynamic team-driven templates. Each member of a team has their own personal view of the site, allowing them to customize the user interface and displayed content.

2.3 Telepresence Module

The telepresence module is comprised of a streaming video layer and a collaborative communications layer. Streaming video from each of the experiment setups provide students with visual and auditory feedback that is necessary for achieving an immersive user environment. Video allows students to take remote readings from analog gauges, and also increases the transparency of the user interface, allowing students to directly experience what the experiment is actually doing. The same streaming video system is also used to display live video of the laboratory environment on the internet, which is important for extending the full laboratory experience to the web domain.

The communications layer serves to connect remote students with the instructor, TAs and with each other. During the course of an experiment a student can access the instructor or TA using a number of web-based communication tools. If synchronous text messaging is insufficient the student can elevate the communication mode to live audio or video chat, although a microphone and webcam are required for these features. More complex communication modes are accessible

∗ Email: emami@utias.utoronto.ca

through application sharing, which allows a group of users to share a common whiteboard, software application, or even an entire desktop session. The communications layer is also integrated with the web interface layer allowing a user's online presence to follow them throughout the portal. Within the portal a user's presence is linked to their online activity, and is broadcast through any instance of their username. Students can therefore determine if the instructor or a TA is available directly from the experiment interface page.

2.4 Application Publishing Module

The application publishing module is an extension of the telecontrol layer that allows students to access laboratory software and computing resources from a centralized application server using RDP. The instructor can define a list of pre-approved applications that students can run, along with a set of permissions for each application (i.e., what the user can do with the application). Software usage is logged to a database so that the instructor can monitor resource utilization.

2.5 Scheduling Module

Scheduling of student access to the remote access experiments is managed through a central scheduling system. The scheduling system consists of a scheduler engine and a central database containing the master schedule for all experiments. To avoid procrastination on the part of the students, access to each experiment is scheduled to ensure that every student has at least two designated 3-hour time slots per experiment. Between scheduled time slots the experiments are also available on a first-come-first-serve basis.

3. eLaboratory Implementation

3.1 Remote Experiment Module

The remote experiment module is implemented using a Target PC deployment with each experimental hardware setup being hosted on a dedicated workstation. Typically a high-end Pentium® III or low-end Pentium® IV is sufficient for running each experiment. A standard Peripheral Component Interconnect (PCI) DAQ board that is used in most of the experiments is a PCIM-DAS1602/16 board from Measurement Computing. This board has 32 digital Input/Ouput (I/O) channels, and a 16 channel Analog-to-Digital (A/D) converter with 16-bit resolution (i.e. 2^{16} discrete voltage values over the measurement range of -5V to 5V), which is equivalent to a voltage resolution of 0.15mV. The DAQ board allows the software interface, running on the Target PC, to control electromechanical actuators and collect data from a variety of analog and digital sensors.

Each Target PC runs Windows Server® 2003 R2 as the core OS and is configured as a Terminal Server. A Terminal Server RDP session is transparent to any software running on the Target PC allowing the GUI for each experiment to be completely independent of the telecontrol layer. Therefore previously developed control applications (written in any language) can be easily reused without the need to program a new interface.

Although a server platform is not required for running the experiments, it provides a much more comprehensive development environment over a client OS such as Windows® XP. A full Terminal Server is also not required, but has significantly more advanced management features than a standalone Remote Desktop installation.

The GUI for each experiment is a multithreaded graphical application written in Visual C++® that communicates with the DAQ boards using a C/C++ Application Programming Interface (API) provided by the DAQ board manufacturer. Visual C++® was chosen for programming the GUI since it is a flexible language that most undergraduate engineering students are experienced with. This was an important consideration in the design of the software interface since most of the software development has been completed by undergraduate students.

Although each experiment has a unique software interface there are a significant number of functions that are common to all experiment GUIs. These common functions were written in Visual C#® and compiled into an assembly, allowing the common source code to be shared and updated without having to recompile each experiment interface. Visual C#® was chosen over Visual C++® for programming the common functions since it is a managed language (the runtime handles memory allocation) and hence is less prone to programming errors. Visual C#® also has a simpler interface for interacting with the core Windows® OS; an important consideration since the common functions rely heavily on platform invocations (calling methods from unmanaged assemblies). An example functionality that is common to all experiments is the automatic mapping of a student's personal document library to a network drive, using the Web-based Distributed Authoring and Versioning (WebDAV) protocol, which allows the application to then save data directly to the student's personal site.

The final and most important component of the software interface is the watchdog. The watchdog must be robust, fault tolerant and capable of recovering from a variety of failures. To that end the watchdog is not a single software component, but rather a collection of distributed interlinked management technologies (e.g., scripts, services, performance counters, etc.)

∗ Email: emami@utias.utoronto.ca

with no single point of failure. Any failure in the software interface or watchdog layer causes the entire experiment to be shutdown.

3.2 Engineering Portal Module

The engineering portal module is an internet portal built on Microsoft® Office® SharePoint® Server (MOSS) 2007. The architecture of the portal is analogous to a corporate intranet with the portal serving as a remote access gateway to hosted laboratory resources.

The core portal software is a collection of ASP.NET Web Applications, Web Services and Collaborative Application Mark-up Language (CAML) templates that provide a media-rich interface to what is essentially a sophisticated database. ASP.NET is a web technology that uses server side code, running against the Microsoft® .NET Framework, to dynamically generate HTML. It enables powerful backend applications to be fronted by interactive web-based user interfaces.

3.3 Telepresence Module

The telepresence module is implemented using a number of enterprise grade video surveillance and collaborative communications tools.

Sony ¼″ color Charge-Coupled Device (CCD) analog dome cameras are used to capture video from the remote experiment setups. Sony ECM-MS907 One Point Stereo Microphones are used to capture audio from the remote experiment setups. The analog audio/video channels feed into a pair of GeoVision GV-1240 video capture cards installed on a set of dedicated video servers. Each GV-1240 capture card converts 16-channels of combined audio and video into a digital stream that can then be served to clients at 30 fps on all channels.

The video servers are dual-core Intel® Pentium® D workstations, each with a video capture card for a total of 32-channels of streaming audio/video. The GeoVision Web Server, provided with the capture cards, is used to stream audio/video to GeoVision LiveX ActiveX® Controls (or Java™ applets, although the ActiveX® Controls are more memory efficient) embedded in the experiment interface pages. Centralizing remote perception for all of the experiments greatly reduces the complexity of managing the streaming audio/video and removes the burden of processing multimedia streams on the individual Target PCs.

The communications layer is comprised of a Session Initiation Protocol (SIP) messaging network that connects students, instructors and TAs in real-time. Users access the communications network using either a web-based Instant Messaging (IM) client or a full SIP client that includes support for Voice over IP (VoIP), video chatting, and shared workspaces. The communications network runs on Microsoft® Live Communications Server 2005 and uses Office® Communicator Web Access as the web-based IM client and Office® Communicator 2005 as the SIP client.

3.4 Application Publishing Module

The application server runs Windows Server® 2003 R2 configured as a Terminal Server. At logon the application server does not start the traditional Windows® desktop (i.e. explorer.exe), instead it runs a custom publishing application, wirtten in Visual C#® that allows the user to start the pre-approved list of applications. When a student logs onto the application server their personal document library is mapped to a network drive using the WebDAV protocol.

Users can access the application server either directly through a remote desktop client (e.g. the Remote Desktop Connection client in Windows®), or through the Remote Desktop Web Connection client that can be embedded in an HTML page. The later method is used to provide students with access to the application server from within the eLaboratory Portal.

Computing cluster support is provided using Microsoft® Cluster Services, which allow a collection of distributed workstations to be interconnected and dynamically load balanced across a large user group.

3.5 Scheduling Module

The schedule engine is a custom service written in Visual C#®. The schedule service wakes up every hour, loads the current schedule from the masterdatabase, logs off any student that are not scheduled, and grants access to students that are scheduled.

For the sake of simplicity the schedule database is an Excel® spreadsheet with entries for each timeslot. A spreadsheet was chosen as the schedule database since it is easy to modify and can be accessed using the generic Object Linking and Embedding Database (OLE-DB) API, although a more power database application such as Access® or SQL Server would have worked equally as well.

∗ Email: emami@utias.utoronto.ca

The scheduler uses the Terminal Services API to log off students that have gone passed their scheduled time slot. Access permissions are controlled using Windows® Security Groups at the domain level. The scheduler removes the previous user from the "*Experiment User*" group in the domain and then adds the next scheduled user to that group. Therefore, access to each experiment is extremely secure and reliable since it is controlled at a domain level using the built-in security functionality of Active Directory®.

3.6 Report Templates

Report templates are used to standardize the presentation of results, minimize time spent on formatting graphs and tables and make the reports easier to read and grade for the TAs. Each template consists of a Word® document with pre-defined fields for each section of the report (e.g., introduction) and a series of embedded Excel® objects for displaying graphs and tables. Each embedded Excel® object is pre-configured to import external data from the data files collected during each experiment. Students select which data file to load for each Excel® object and at the click of a button the data is automatically imported into a pre-formatted figure. This feature saves students a significant amount of time in formatting their plots, particularly for experiments that generate many datasets, without limiting their freedom in producing creative scientific writing

3.7 Network Topology

The eLaboratory infrastructure is composed of five core servers; namely the Domain Controller (DC), Portal Server, Real-Time Communications (RTC) Server, Video Server, and Application Server. Each remotely-accessible experiment also has a dedicated server, called the Target PC. To ensure complete hardware independence of the implementation, all server applications are run on virtual machines hosted on the physical hardware. Virtual machines use a combination of hardware pass-through and virtualization technologies to run a guest OS on top of the host server. Virtual machines can therefore be easily migrated between different physical servers.

The eLaboratory network runs under a Microsoft® Active Directory® domain with each server running Windows Server® 2003 R2. The eLaboratory Portal is based on modular components of the Windows Server System® and includes Microsoft® Office® SharePoint® Server 2007 (MOSS), Office® Live Communications Server 2005 (LCS), Office® Communicator Web Access (CWA), and SQL Server 2005.

Each physical server supports one or more virtual machines using Microsoft® Virtual Server 2005 R2. The network is hosted on a pair of stacked 3Com® SuperStack® 3 4400-series switches (100Base-T) with a high-bandwidth uplink directly to the University network backbone via fiber optics. Inter-server communications is isolated on a dedicated 3Com® SuperStack® 3 3800-series switch (1000Base-T) to ensure a secure and highly available channel for critical network traffic. Hardware and software specifications for each server are summarized in Table 1.

TABLE 1

HARDWARE AND SOFTWARE REQUIREMENTS FOR THE ELABORATORY PORTAL.

Server	Processor	RAM	NIC	HDD	Function
Target PCs	Pentium® IV (x86) 2.0MHz	256 MB	100 Mbps	10 GB	-remote desktop -user interface
Application	Dual Xeon® (x64) 2.8 GHz	6 GB	1 Gbps	47 GB	-remote desktop -software resources
Communications	Dual Xeon® (x64) 2.8 GHz	4 GB	1 Gbps	47 GB	-communications -IM web client
Portal	Dual Xeon® (x64) 3.0 GHz	4 GB	1 Gbps	500 GB	-engineering portal -backend database
Domain Controller	Dual Pentium® III (x86) 1.1 GHz	2 GB	1 Gbps	47 GB	-domain controller -firewall and VPN
Video	Pentium® D (x86) 3.2 GHz	1 GB	1 Gbps	250 GB	- video capture -streaming server

∗ Email: emami@utias.utoronto.ca

4. Evaluation

The eLaboratory architecture was used for two instructional laboratory courses (AER303 and AER304) for third-year undergraduate Aerospace students in the Engineering Science program at the University of Toronto. A seven-question survey was used to evaluate students' perceptions of the effectiveness of the eLaboratory. The survey questions were based on a similar survey used by Lindsay *et al.* [21]. Students were also asked to complete the VARK [24] learning style catalyst, which measures students' cognitive learning style preferences across four categories: namely visual, aural, read-write, and kinesthetic.

The survey responses were converted from nominal categorical data to numerically encoded data (i.e., each categorical response to a particular question was encoded as a different positive integer), so that the results could be tabulated and cross-correlated using the MATLAB® (R2007a) Statistics Toolbox. Several correlations are observed. Students who found the course intellectually stimulating also tended to enjoy the course more than those who did not. Clearly, interest in a course can lead to improved enjoyment, which may lead to improved educational outcomes. Interestingly, students, who thought the data they collected were accurate, also tended to enjoy the course more than those who did not. In addition, the perception of data accuracy is related to the preferred access modality. Students who felt their data were accurate also tended to prefer remote experimentation or show no preference at all.

In both courses the majority of students either preferred remote access or showed no preference. The percentage of those who preferred proximal access remains almost the same in both courses. However, the reasons for this preference as indicated by the students are different. There were concerns with the transparency of the user interface, with the most prevalent comment being that students did not understand what the apparatus was doing when a particular button on the user interface was clicked. Concerns with the transparency of the experiment interface tended to correlate (confidence level higher than 95%) with the perception that the data collected were somehow incorrect or inaccurate. These results support the notion introduced by Benmohamed *et al.* [25], which establishes that a remote experiment interface must be designed for specific pedagogical activities to provide learning support, but that a balance must be struck to maintain transparency.

Another common concern with the remote access was the lack of physical interaction with the equipment, with one student commenting "*Proximal, I want to smell the oil.*" Interestingly, the only two students who performed the experiments both remotely and proximally (using a workstation in the laboratory) showed no preference for either modality. This may indicate that the preference for proximal interaction is a result of students' preconceived notion of what constitutes a "laboratory." More details on the analysis of students' feedback are discussed in [26].

Qualitative feedback collected from the TAs indicated that there was a fundamental problem with the remote communications tools. The TAs felt that synchronous text messaging (i.e. instant messaging) was not sufficient for resolving problems with the students. Although video messaging is the obvious solution to this deficiencey, not all students have access to a webcam, which makes video chatting limited in its application. This is confirmed by the fact that despite support for video chatting in the eLaboratory architecture, no student used this communications mode.

Several typical positive responses to the remote access modality include:

- "*simplified access/communication*"
- "*easier on time and travel*"
- "*it makes the lab more straightforward*"

Several typical negative responses to the remote access modality include:

- "*a sense of lack of motivation to do it at home*"
- "*took out "hands-on" experience, made it difficult to visualize…*"
- "*involving new technology inevitably may let more things go wrong*"

5. Conclusions

Remote access laboratories are gradually emerging in the engineering curricula as platforms for inter-university collaborations aimed at establishing global distance education. Nonetheless, more needs to be addressed in the literature about unified and consistent frameworks for their implementations as well as their pedagogical implications. This paper presented an architecture for remotely-accessible laboratories, which is based on a convergence of remote access technologies and collaboration-based eLearning, and also discussed the implementation of such a framework. The generic and modular design makes the eLaboratory architecture applicable to major engineering disciplines, and allows new installations to be rapidly deployed and reconfigured.

An evaluation of the eLaboratory architecture by students indicated that they are generally receptive to new technologies introduced into their learning process. However, abstracting students from the physical hardware shifts the focus from the experimental setup to the user interface and remote technologies. Hence, in the case of hardware failure students tend to

∗ Email: emami@utias.utoronto.ca

associate the failure with the remote aspect of the experiment, rather than with the physical hardware. In addition, there is a general perception that proximal experimentation would allow these failures to be easily corrected. The major areas of concern that were highlighted in the evaluation were an overall lack of transparency in the user interface and insufficient communications between the students and TAs. Both of these concerns are related to the telepresence aspect of the user experience. Future works should attempt to address these concerns by increasing the transparency of the user interface. Also, network latency and bandwidth are major technological bottlenecks for remote access laboratories, and therefore techniques of enhancing audio-visual streaming and RDP communication within these limits must be continually investigated.

ACKNOWLEDGEMENT

This work was funded in part by the Kenneth M. Molson Foundation and the University of Toronto Provost's Office.

REFERENCES

[1] J. Puustjarvi, "Integrating e-learning systems," in *Proc. IASTED International Conference on Web-based Education*, Innsbruck, Austria, 2004, pp. 417-421.
[2] M. Nichols, "A theory for eLearning," *Educational Technology and Society,* vol. 6, pp. 417-421, 2003.
[3] F. Davoli, G. Spanò, S. Vignola, and S. Zappatore, "LABNET: Towards Remote Laboratories With Unified Access," *IEEE Transactions on Instrumentation and Measurement,* vol. 55, pp. 1551-1558, 2006.
[4] E. Voelki, L. F. Allard, J. Bruley, and D. B. Williams, "Undergraduate TEM Instruction by Telepresence Microscopy Over the Internet," *Journal of Microscopy,* vol. 187, pp. 139-142, 1997.
[5] 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 Transactions on Education,* vol. 42, pp. 174-179, 1999.
[6] C. C. Ko, B. M. Chen, J. Chen, Y. Zhuang, and K. C. Tan, "Development of a web-based laboratory for control experiments on a coupled tank apparatus," *IEEE Transactions on Education,* vol. 44, pp. 76-86, 2001.
[7] A. Ferrero, S. Salicone, C. Bonora, and M. Parmigiani, "ReMLab: A java-based remote, didactic measurement laboratory," *IEEE Transactions on Instrumentation and Measurement,* vol. 52, pp. 710-715, 2003.
[8] G.-W. Chang, Z.-M. Yeh, H.-M. Chang, and S.-Y. Pan, "Teaching photonics laboratory using remote-control web technologies," *IEEE Transactions on Education,* vol. 48, pp. 642-651, 2005.
[9] X. Tang and K. Li, "Developing a efficient remote lab environment for online IDS courses," in *Proc. 2005 ASEE Annual Conference & Exposition*, Portland, OR, USA, 2005.
[10] S. Wirgau, A. Gupta, and V. Matzen, "Internet-Enabled Remote Observation and Control of a Shake Table Experiment," *Journal of Computing in Civil Engineering,* vol. 20, pp. 271-280, August 2006 2006.
[11] R. Wirz, R. Marín, J. F. Ruzafa, and P. J. Sanz, "Remote Programming of Multirobot Systems within the UPC-UJI Telelaboratories: System Architecture and Agent-Based Mutlirobot Control," in *Proc. IEEE Workshop on Distributed Intelligent Systems*, Prague, Czech Republic, 2006.
[12] J. Plaffman, "Open source solutions: Moodle," *Learning & Leading with Technology,* vol. 33, pp. 42-44, 2005.
[13] S. Rapuano and F. Zoino, "A Learning Management System Including Laboratory Experiments on Measurement Instrumentation," *IEEE Transactions on Instrumentation and Measurement,* vol. 55, pp. 1757-1766, October 2006 2006.
[14] M. Casini, D. Prattichizzo, and A. Vicino, "The automatic control telelab: a user-friendly interface for distance learning," *IEEE Transactions on Education,* vol. 46, 2003.
[15] D. Krehbiel, R. Zerger, and J. K. Piper, "A Remote-Access LabVIEW-based Laboratory for Environmental and Ecological Science," *International Journal of Engineering Education,* vol. 19, pp. 495-502, 2003.
[16] H.-Y. Lu, X.-Q. Wu, Z.-H. Liu, D.-C. Bu, S.-Y. Tan, and L.-Y. Li, "Realization of Remote Electromagnetism Experimental Teaching System Based on LabVIEW," *Research and Exploration in Laboratory,* vol. 25, pp. 634-637, June 2006 2006.
[17] M. M. Albu, K. E. Holbert, G. T. Heydt, S. D. Grigorescu, and V. Trusca, "Embedding remote experimentation in power engineering education," *IEEE Transactions on Education,* vol. 19, pp. 139-143, 2004.
[18] W. E. Dixon, D. M. Dawson, B. T. Costic, and M. S. d. Queiroz, "A MATLAB-based control systems laboratory experience for undergraduate students: toward standardization and shared resources," *IEEE Transactions on Education,* vol. 45, pp. 218-226, 2002.
[19] J. Sánchez, S. Dormido, R. Pastor, and F. Morilla, "A Java/Matlab-Based Environment for Remote Control System Laboratories: Illustrated With an Inverted Pendulum," *IEEE Transactions on Education,* vol. 47, pp. 321-329, August 2004 2004.
[20] J. E. Corter, J. V. Nickerson, S. K. Esche, and C. Chassapis, "Remote Versus Hands-On Labs: A Comparative Study," in *Proc. 34th ASEE/IEEE Frontiers in Education Conference*, Savannah, GA, USA, 2004.
[21] E. D. Lindsay and M. C. Good, "Effects of Laboratory Access Modes Upon Learning Outcomes," *IEEE Transactions on Education,* vol. 48, pp. 619-631, November 2005 2005.
[22] M. Helander and M. R. Emami, "eLearning for Engineering," in *Proc. 9th International Conference on Engineering Education*, San Juan, Puerto Rico, 2006.
[23] M. R. Emami and M. Helander, "eDesign," in *The 3rd CDEN/RCCI International Design Conference on Education, Innovation, and Practice in Engineering Design* Toronto, Canada, 2006.
[24] N. D. Fleming and C. Mills, "Not another inventory, rather a catalyst for reflection," *To Improve the Academy,* vol. 11, pp. 137-143, 1992.
[25] H. Benmohamed, A. Lelevé, and P. Prévot, "Generic Framework for Remote Laboratory Integration," in *Proc. 6th Annual International Conference on Information Technology Based Higher Education and Training*, Juan Dolio, Dominican Republic, 2005.
[26] M.G. Helander and M. R. Emami, "eLearning for Engineering Education," Submitted in *International Journal of Engineering Education*, August 2007.

∗ Email: emami@utias.utoronto.ca