INVITED PAPER

The iLab Shared Architecture: A Web Services Infrastructure to Build Communities of Internet Accessible Laboratories

If you can't come to the lab ... the lab will come to you!—Jesus del Alamo

By V. Judson Harward, Jesus A. del Alamo, *Fellow IEEE*, Steven R. Lerman, Philip H. Bailey, Joel Carpenter, Kimberley DeLong, Chris Felknor, James Hardison, *Member IEEE*, Bryant Harrison, Imad Jabbour, *Member IEEE*, Phillip D. Long, Tingting Mao, Loai Naamani, Jedidiah Northridge, Mark Schulz, Daniel Talavera, Charuleka Varadharajan, Shaomin Wang, Karim Yehia, Rabih Zbib, and David Zych

ABSTRACT | The Massachusetts Institute of Technology's iLab project has developed a distributed software toolkit and middleware service infrastructure to support Internet-accessible laboratories and promote their sharing among schools and universities on a worldwide scale. The project starts with the assumption that the faculty teaching with online labs and the faculty or academic departments that provide those labs are acting in two roles with different goals and concerns. The iLab architecture focuses on fast platform-independent lab development, scalable access for students, and efficient management for lab providers while preserving the autonomy of the faculty actually teaching the students. Over the past two years, the iLab architecture has been adopted by an increasing number of partner universities in Europe, Australia, Africa, Asia, and the United States. The iLab project has demonstrated that online laboratory use can scale to thousands of students dispersed on several continents.

KEYWORDS | Educational technology; engineering education; Internet; laboratories

I. INTRODUCTION

The concept of remote access to laboratory equipment arises naturally from telemetry and the well-established trend of using standard computers, usually PCs, to control and record data from local lab apparatus. As scientific

---

Manuscript received November 5, 2007. This work was supported in part by the Microsoft Corporation through iCampus, (the MIT-Microsoft Alliance), by the Carnegie Corporation of New York, by the National Science Foundation under award 0702753, by the Singapore-MIT Alliance, by the Singapore-MIT Alliance for Research and Technology, and by MIT Alumni Funds (Classes of '51, '55, '60, and '72), as well as by National Instruments, Agilent Technologies, AMD, Hewlett-Packard, and Intel by equipment donations.

V. J. Harward, P. H. Bailey, K. DeLong, C. Felknor, J. Hardison, B. Harrison, and **C. Varadharajan** are with the Center for Educational Computing Initiatives, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: jud@mit.edu; pbailey@mit.edu; kirky@mit.edu; felknor@mit.edu; hardison@mit.edu; bryant_h@mit.edu; charuv@mit.edu).

J. A. del Alamo is with the Electrical Engineering and Computer Science Department, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: alamo@mit.edu).

S. R. Lerman is with Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: lerman@mit.edu).

J. Carpenter is with the University of Queensland, Coorparoo, Qld. 4151, Australia (e-mail: J.carpenter@uq.edu.au).

I. Jabbour and **S. Wang** are with Oracle, Inc., Belmont, CA 94002 USA (e-mail: ijabbour@alum.mit.edu; smwang@alum.mit.edu).

P. D. Long is with the Office of Educational Innovation and Technology, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: longpd@mit.edu).

T. Mao is with the Department of Civil and Environmental Engineering, Massachusetts Institute of Technology, Cambridge, MA, 02139 USA (e-mail: tmao@mit.edu).

L. Naamani is with the Department of Civil and Environmental Engineering, Massachusetts Institute of Technology, Boston, MA 02116 USA (e-mail: loai@mit.edu).

J. Northridge is with ClearPoint Metrics, Belmont, MA 02478 USA (e-mail: jedidiah@alum.mit.edu).

M. Schulz is with the University of Queensland, Brisbane, Qld. 4072, Australia (e-mail: m.schulz@uq.edu.au).

D. Talavera is with the Naval Nuclear Propulsion Program, Alexandria, VA 22304 USA (e-mail: calitalv@mit.edu).

K. Yehia is with Deutsche Bank, New York, NY 10011 USA (e-mail: kyehia@alum.mit.edu).

R. Zbib is with the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: rabih@mit.edu).

D. Zych is with the National Center for Supercomputing Applications, University of Illinois, Champaign, IL 61821 USA (e-mail: dzych@mit.edu).

Digital Object Identifier: 10.1109/JPROC.2008.921607

instrumentation and experimentation has become more expensive and distance education has become more common, Internet-accessible labs no longer appear novel [1], [2]. Most of these efforts, however, have been ad hoc systems that are closely tailored to the requirements of a particular online lab.

In contrast, the iLab Shared Architecture (ISA) is a Web service infrastructure that has been developed at the Massachusetts Institute of Technology (MIT) to provide a unifying software framework that can support access to a wide variety of online laboratories. Users and the online laboratories can be globally distributed across an arbitrary number of locations linked only by the Internet. Users access these remote laboratories through single sign-on and a simple standard administrative interface.

This paper starts by describing the origin of the iLab project. It then states the project's goals and how they differ from those of related work. A central section describes the software architecture as it applies to two major types of online experiments: batched, in which the entire experiment is defined before execution starts, and interactive, in which the user can observe and modify the course of the experiment in real time. In each case, we discuss how the ISA provides experiments with generic services such as authentication and result storage. We also describe standard approaches to the design and implementation of the lab client that provides the interface to the experiment seen by users and the lab server that controls the actual execution of the experiment. This paper then turns to examine how the use of online labs affects pedagogy and what factors contribute to the creation of effective online labs.

The ISA framework enables the sharing of online labs, and this paper explores those conditions that favor the adoption of the iLab technology and such sharing of lab resources between institutions. It concludes by describing a potential new organization, the iLab Consortium, intended to foster the growth of the technology and to set priorities as the iLab community grows.

II. THE ORIGIN OF THE iLAB PROJECT

The iLab project was started at MIT in 1998 by one of the authors (J. A. del Alamo). It was several years before the project acquired its final name. The initial inspiration for the first iLab came from the frustration that MIT's courses on semiconductor devices did not contain a laboratory component. Traditionally, students in these courses were exposed only to theoretical device models presented in lectures and course texts. At the same time, an Agilent 4155B semiconductor parameter analyzer, an expensive piece of equipment bought under a research contract, was sitting in a graduate research lab with spare capacity available. While the underutilization of the Agilent equipment seemed to provide an opportunity to have this tool also used in education, there was no way to accommodate the students taking courses using a single piece of equipment in the crowded research lab.

A small initial grant from the Microsoft Corporation allowed del Alamo to explore the potential of remote access to the 4155B. He hired an undergraduate student who wrote a Java applet that enabled students using a standard Web browser to submit descriptions of semiconductor device characterization routines for execution by a server connected directly to the Agilent instrument. Students in an upper level electrical engineering course were the first to try this system, Microelectronics WebLab, in the fall of 1998. By the following spring, the hardware and software combination had proved its reliability to the point that an undergraduate class of nearly 100 students employed the online lab for an assignment [3], [4].

Late in 1999, del Alamo persuaded colleagues from a number of engineering departments to apply jointly to the newly formed research partnership between MIT and Microsoft known as the iCampus project. The goal of this proposal was the creation of a diverse set of iLabs (ranging from a flagpole instrumented with accelerometers to a remotely controlled heat exchanger) to further explore the potential of online labs in undergraduate education. In this next phase of the project, each team developed its Web-accessible lab independently using a wide variety of software techniques.

During this period, development of the Microelectronics WebLab continued [5], [6], [32]. The addition of a switching matrix allowed the system to host multiple semiconductor devices for characterization. This allowed redundant devices to enhance reliability while it also permitted users to test different devices in different courses simultaneously. Greater confidence in the reliability and scalability of the system led to a significant expansion in the use of the lab not only inside MIT but also at other institutions. The first international use of the system came in the fall of 2000 in the context of a collaboration between MIT and two universities in Singapore known as the Singapore-MIT Alliance (SMA). This collaboration was to continue throughout the life of SMA. International use expanded over the following years, culminating in the spring of 2003 in the largest course supported by this system to date: an undergraduate subject from Chalmers University in Sweden with 350 students [7]. More than 5400 students from nine countries on four continents have now used the Microelectronics WebLab for graded project assignments in formal classes (Fig. 1).

The initial Java-based architecture of the Microelectronics WebLab was reliable and scalable enough to allow us to explore key issues involved in the development and sharing of online laboratories on a worldwide scale. In this phase of the project, we learned that developing new labs from scratch required considerable effort and that the domain specialist, only under rare circumstances a software engineer, had to play a key role. This represented

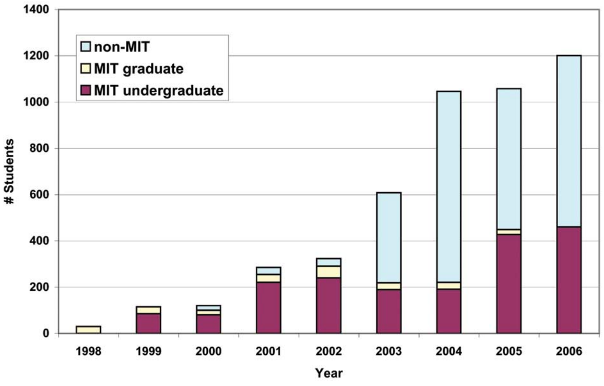


Fig. 1. ***Student use of the Microelectronics WebLab in formal graduate and undergraduate courses. All these students were asked to carry out a credit bearing assignment that contributed to the final grade in the course.***

a significant technical challenge. We also learned that managing a large number of students for a course taught at another institution imposed a sizeable load on the lab manager. This burden constituted a disincentive to lab sharing. These extensive educational experiments became a crucial formative experience that played a pivotal role in the design of the iLab architecture.

Previously in 2001, H. Abelson of MIT's Electrical Engineering and Computer Science Department and D. Mitchell of Microsoft, who were both involved in the leadership of the iCampus project, suggested that the various iLab researchers might progress more quickly if they based their online labs on a shared infrastructure. Abelson and Mitchell also made the crucial suggestion that the infrastructure be built on top of the recently introduced technology of Web services. MIT's Center for Educational Computing Initiatives joined the effort to lead the development of the resulting middleware known as the iLab Shared Architecture. Over the following few years, this enlarged project team developed a set of specifications and a reference implementation for the iLab batched architecture.

The first online lab developed under the ISA was a new version of the Microelectronics WebLab. Deployment took place in the fall of 2004, and the original Java version of this lab was discontinued the year after. Since then, multiple laboratories have been developed under the ISA, both at MIT and elsewhere. More recently, the interactive version of the ISA has also been completed and deployed. Just as the batched architecture grew out of lessons learned from the early Microelectronics WebLab, the interactive architecture benefited greatly from the extensive experience accrued during the development of the Heat Exchanger WebLab by C. Colton's team at MIT [8], [9] and the Polymer Crystallization WebLab developed by G. Rutledge's team at MIT [10] in the early days of the iCampus project.

III. iLAB PROJECT GOALS

A. The Goals of the iLab Project

Our experience with the early WebLabs described above and particularly with the Microelectronics WebLab shaped our belief that the new shared architecture should facilitate the scalability of both lab development and user management. We hoped that reducing costs along these two dimensions would favor our overall goal of encouraging the development and global sharing of online labs.

In the fall of 2002, the Shared Architecture team started the process of gathering more detailed requirements for the new architecture. This involved examining both the various categories of end users and the various potential types of laboratory equipment and experiment protocols that the architecture would eventually need to serve. From this process emerged a set of principles that continues to lead the design and development of the ISA today.

- The main constituency for the ISA would be teaching faculty and students. From the beginning it was clear that the architecture could also be useful for research but that would be an added benefit, not the initial goal.
- While the ISA could provide access to simulations running either locally on the student's computer or centrally on a server, its primary goal was to provide students access to real laboratory equipment.
- The ISA had to implement a highly scalable environment federated in such a way that it could serve a potentially unlimited number of users and online laboratories. The access to a particular piece of laboratory equipment should only be limited by the duty cycle of the apparatus or intentional constraints imposed by the lab manager, not the configuration and overhead of the middleware.
- The ISA must separate the responsibility for delivering an online laboratory experience from that of managing the students who use it. Lab developers generous enough to share their laboratory equipment should not be burdened with administering the usage of students they are not teaching.
- The ISA should foster easy use and administration for all participants in the system: lab providers, system administrators, instructors, and students.
- The ISA must permit laboratory equipment to be accessed through multiple interfaces adapted to the different pedagogical levels and computing environments of users.

B. iLabs Contrasted With Other Technologies

Many projects are currently trying to provide students and researchers access to online laboratory experiences. The iLab project is unusual in its approach because it is not domain specific. It attempts to provide a unifying context and middleware to support online laboratories from a wide variety of fields. As such, it differs from and complements a large number of projects focused on particular domains. A comparison with some of these efforts will help clarify iLab's goals and approach.

1) Simulations: Educators have debated the corresponding benefits of actual online laboratories versus simulations [11]–[13]. An online simulation, once correctly implemented, has a durability and negligible cost of operation that offers clear advantages over a corresponding true lab experiment. The nanoHUB Web site[1] exemplifies the increasing sophistication of such Web-accessible simulations and their great value for engineering education [14].

We have found that students' experience in using online labs differs from that of using simulations largely in their reaction to analyzing the noisier and more complex data from the online labs. In the case of one MIT iLab (the Microelectronics WebLab [3]), an experimental version of the online lab allows students to compare actual semiconductor characterizations with simulated theory-based models. The difference between real data and models drives the next cycle of analysis and understanding. While the ISA can treat a real lab and a simulation similarly, true labs often possess issues of access and equipment control that simulations lack. The iLab architecture has been developed and optimized to provide students secure and efficient access to true lab equipment by taking into account these access and control issues.

2) Instrument Specific Software: The vendors of sophisticated automated lab instruments frequently market their devices with operating and analysis software designed to execute on a computer interfaced to the instrument. This not only allows the vendor to expose the full feature set of their apparatus to the user but also often provides them with a separate revenue stream. Such vendor software rarely supports remote access. Some projects have used virtualization and application streaming to provide remote user access to the vendor application window on the lab server controlling such high-end equipment. This approach resembles a finer grained version of using a standard remote desktop. For example, the CASPiE Project[2] uses the Citrix presentation server to provide students access to a range of sophisticated instruments for analytical chemistry including gas and liquid chromatography, mass spectroscopy, and Fourier transform infrared/Raman spectroscopy. In such an approach, the lab provider typically assumes responsibility for managing the remote user accounts and storing, at least initially, the resulting data because the remote user is using the proprietary vendor software on the lab provider's systems. A user typically has to log in separately to each instrument network because they will require different credentials.

The goals of the ISA are compatible and complementary to such an approach. The iLab middleware layer provides single sign-on for students to an arbitrary number of online labs, frees the lab provider from the responsibility for managing student accounts, and allows students to schedule access to heavily used lab equipment in advance. We have investigated an approach where the iLab middleware interposes between the student's authentication and access to the application server running the instrument specific vendor-supplied software. As long as the application service allows this interposition, the ISA can provide its standard benefits while still permitting the use of the vendor-supplied software.

3) Proprietary Toolkits—The LabVIEW Model: Since the range of potential online labs is unlimited, it is difficult to design a set of standard interfaces to them. National Instruments' LabVIEW product[3] [15], [16] has addressed this problem by creating an application server that runs on a computer directly connected to the laboratory apparatus. This application server employs a dataflow programming model and a sophisticated graphic user interface package that provides ready-to-use components resembling standard lab equipment such as knob and switch controls, meters, and strip charts. The lab developer uses the LabVIEW programming environment in combination with data acquisition cards to build an experiment controller on the lab server. A user who is logged into the lab server can then control and monitor the pieces of lab equipment interfaced to that system. Using the rich LabVIEW environment, a lab provider can develop appropriate interfaces for almost any computer controlled experiment in a fraction of the time that would be required if they employed a standard graphical user interface toolkit (e.g., Java Swing).

The LabVIEW software also provides remote access to the controlling lab server through a proprietary browser plug-in and web server associated with the LabVIEW runtime environment. By default, the LabVIEW runtime will allow any remote user who knows the URL of the lab display page to view the progress of the lab. Control of the experiment goes to the first user to request it. This approach suffers, however, from the same management drawbacks as the previous category of remote labs. Either lab providers accept unrestricted first come-first served access or they assume responsibility for managing access to the lab server from within the LabVIEW environment.

The iLab middleware and the LabVIEW environment can form a powerful combination to overcome these drawbacks. LabVIEW offers a flexible user interface toolkit adapted to controlling lab environments but not to administrative processes like authenticating users and scheduling experiments. The iLab middleware can provide users secure and uniform access to labs whether or not they are implemented with LabVIEW. The process of

---

[1] http://www.nanohub.org.

[2] http://www.purdue.edu/dp/caspie/index.html.

[3] http://www.ni.com/labview.

publishing" an existing LabVIEW experiment has been standardized in the current iLab environment so that it can be achieved on the scale of a few hours.

4) Grid Computing: Grid computing is a growing force in the scientific community. The goals of the grid are to provide vast computing resources to researchers via the distributed system integration of computation and data storage resources [17], [18]. Core grid technologies provide tools that allow members of a grid community to utilize surplus computing power, data storage capabilities, and data collection services that are owned by other members of the community. These resources expand the amount of accessible data and provide new analytical tools to process it.

Both the ISA and the grid support remote access to instrumentation. The grid approach focuses on defining architectures for the "virtualization" of instruments. Virtualization provides abstractions of the actual instruments, related control and data structures, and a suite of authorization and access procedures. These architectures are a more complex extension of the goals of the Interchangeable Virtual Instrument Foundation,[4] a consortium of test equipment manufactures and users. Grid-based instrument virtualization is an active area of research and development. Major initiatives include the GRIDCC's[5] Virtual Instrument Grid Service [19] and the Common Instrument Middleware Architecture (CIMA) [20], [21] supported by the National Science Foundation Middleware Initiative.

The ISA differs from the grid approach in making users rather than resources the focus of the architecture. Since the ISA optimizes the execution of remote experiments, which could be implemented as grid resources, these two approaches are once again complementary rather than conflicting. The iLab team expects to develop a generic bridge to grid-based experiments using one of the instrument virtualization architectures, most probably CIMA.

IV. THE iLAB SHARED ARCHITECTURE

From the perspective of the ISA, online experiments fall into three broad categories.

1) *Batched experiments* are those in which the entire course of the experiment can be specified before the experiment begins. MIT's Microelectronics WebLab [3], [22] provides an example. Through WebLab, students can characterize a variety of semiconductor devices by preparing a test protocol. This is accomplished by using a graphical editor to set parameters before the semiconductor characterization executes. Experiment execution takes place in machine time.
2) *Interactive experiments* are those in which the user monitors and controls one or more aspects of the experiment during its execution. In MIT's TeachSpin Lab,[6] students can dynamically change the frequency and amplitude of an alternating current fed into a Helmholtz coil. The experiment permits them to observe and measure the motion of a magnet suspended in the center of the coil. Experiment execution takes place in human time.
3) *Sensor experiments* are those in which users monitor or analyze real-time data streams without influencing the phenomena being measured. MIT's instrumented flagpole is a simple example [23].

Each category of experiment requires a different mix of shared services. Since the user completely specifies a batched experiment before execution of the experiment begins, the user need not be online when the experiment is performed but instead can retrieve the results later. This implies that batched experiments should generally be queued for execution in a way that maximizes the efficient use of the lab server rather than scheduled to maximize the convenience of the user.

Since the user can control and alter at least some of the inputs of an interactive experiment while it executes, he or she must be online when the experiment runs. If an experiment takes more than a few minutes, students and faculty will normally demand that experiments be scheduled so that students will not waste time waiting for their turn at the apparatus.

A sensor "experiment" often requires the analysis not just of real-time data but also of past sensor data. Such sensor experiments usually require an associated data archive that the user can search for events of interest or use as a source for statistical studies.

The current version of the ISA supports the first two categories of experiments, batched and interactive.

A. The Role of Web Services

The design requirements for the ISA strongly favored the use of Web services as the communication framework for the ISA middleware. Students at one institution must be able to use a lab housed at a second institution. This requires an architecture that supports both lab-side services (e.g., the online lab itself) and client or student-side services (authentication and authorization, class management, student data storage for experiment specifications and results as well as user preferences). The lab-side services may need to run on a different hardware and software platform than the client-side student software. The lab-side institution may enforce different networking policies (e.g., firewalls, directory, and e-mail services) than the client-side institution. The transparency of Web services makes this technology an obvious choice to integrate the iLab distributed application framework.

---

[4] http://www.ivifoundation.org.

[5] http://www.gridcc.org.

[6] http://ni-ilabs.mit.edu/FOD/ForceOnADipole.htm.

In addition, modern lab equipment is frequently interfaced to and controlled by a computer even when remote access is not envisioned. Such existing labs usually possess a large legacy code-base to manage the lab equipment or to analyze and display results. The loose coupling of Web services makes it easier to reuse such legacy code in a second-generation implementation based on the ISA.

B. The ISA Batched Middleware

The ISA batched architecture [24] in some ways resembles the typical three-tier Web business architecture (Fig. 2).

1) The first tier is the student's client application that usually runs as an applet or as a downloaded application on the student's workstation.
2) The middle tier, called the service broker, provides the shared common services. It is backed by a standard relational database such as SQL Server or MySQL. The student's client communicates solely with the service broker, which forwards experiment specifications to the final third tier that includes the lab equipment. Unlike the standard three-tier Web architecture in which the middle tier resides on the business rather than the client side of the network, the service broker normally resides on a server at the student's institution. If a university is willing to provide accounts for users from other institutions, however, the architecture allows the service broker to run on a separate campus from the client; in fact, it can be collocated with the lab itself.
3) The third tier is the lab server, which interfaces with the instruments that execute the specified experiments. The lab server notifies the service broker when the results are ready to be retrieved.

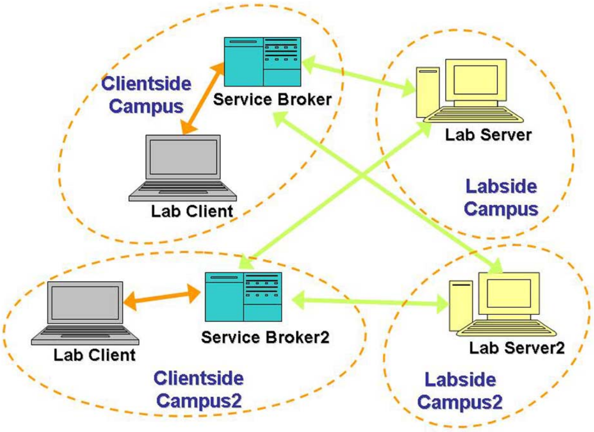


Fig. 2. *The topology of the iLab architecture for batched experiments.*

In the iLab batched architecture, the student client and the lab server both represent the domain- and lab-dependent software modules. The service broker is completely generic code and can interoperate with any combination of client and lab server that implement the appropriate interfaces expressed in terms of Web service Simple Object Access Protocol calls defined in WSDL.

A student starts a session by logging on to the service broker using a standard web browser. Once the student chooses the experiment to execute, the client is launched and communicates with the service broker using the client to service broker Web service. This interface allows the client to transmit to the service broker the description of the experiment to be executed. The service broker stores a copy of the experiment specification before forwarding it on to the lab server via a second service broker to lab server Web service.

The lab server knows nothing about the students using the system, and it only stores experiment specifications and results temporarily. It is the service broker that authenticates students, checks on their authorization to contact a particular lab server, accepts an experiment specification from the student's client, and waits to retrieve the result once the experiment completes. The experiment specification and results are stored on the service broker under the student's account. Thus all the resources consumed by a student, except for the runtime resources required to execute the experiment, can be drawn from a service broker, usually located at the student's institution.

There must be a degree of trust between the lab server and the service broker, first and foremost because the service broker authenticates and vouches for student users. The service broker also indicates the student's level of access to the lab server by forwarding a string key known as the *effective group* when it submits an experiment specification. The lab server does not know on which student's behalf it is executing an experiment. It only knows the requesting service broker and the effective group associated with the request. This allows lab providers to grant different levels of access to different effective groups on multiple service brokers, but it delegates to the service brokers all decisions about the assignment of students or staff to the various effective groups.

Conversely, the service broker knows nothing about the domain-dependent nature of the experiments. When the student first launches an experiment, it forwards an opaque object from the lab server to the student's client describing the current lab configuration. When the student submits an experiment specification, it is forwarded to the lab server as another opaque object, and the results are returned as a third one. The only part of an experiment that the service broker understands is a metadata description of the experiment that can be used to search for and retrieve old experiments. This metadata contains fields common to all experiments, such as the lab server ID and the effective group. We assume that the service broker at one institution may give its students access to lab servers from multiple institutions, and conversely a lab server may receive experiment specifications from service brokers at many institutions.

In the batched experiment architecture, the student's workstation never contacts the lab server directly. We can maintain this strict discipline because a batched experiment requires so little communication between the client and the lab server. Conceptually, the execution of an experiment requires a single round trip over the network, although the actual Web service protocol is more complicated.

C. The Structure of Batched Clients and Lab Servers

In the batched architecture, messages between a lab client and server, which are lab specific by nature, must be transmitted through the generic channels of the service broker. As such, batched-lab development involves the design of three major elements—the lab client, the lab server, and a lab client/server communication framework (LC/SCF) [22]. The goal of the LC/SCF is to encode the lab-specific information that is relayed between lab clients and servers using generic mechanisms. It is also where typical batched-lab development begins. This information usually falls into three sets:

1) the initial setup of a lab and the resources that are available;
2) the parameters defining a particular experiment;
3) the results from the experiment.

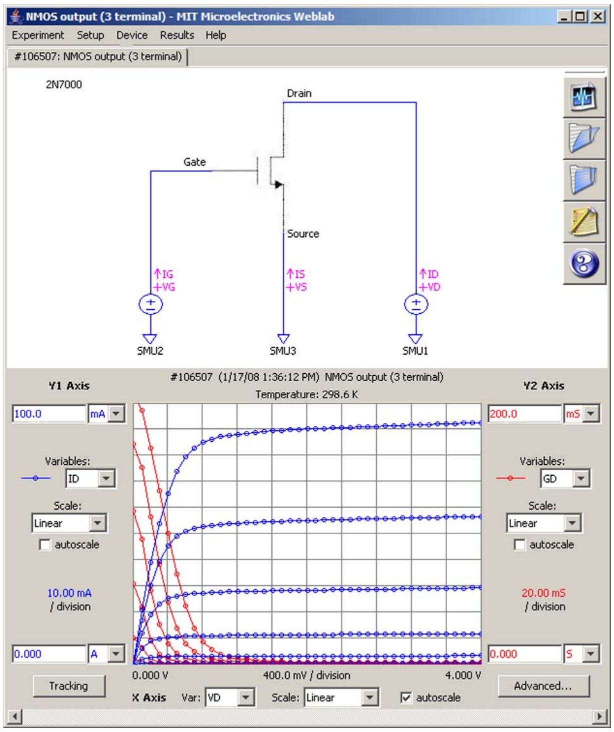
The LC/SCF forms the essence of a particular lab as it defines the parameters and form of that lab's input and output. While the ISA permits the LC/SCF to be expressed in any text format, XML documents are an ideal vehicle for this communication because they can encode specific typed values, e.g., floating-point values, with additional contextual information while being transmitted as plain text. Using XML, the service broker only needs to be able to pass text strings in order to provide the communication of typed data records between any lab client/server pair.

Since the lab client allows the user to set experiment parameters for a given execution and presents the results from a completed experiment, the client must be able to produce and interpret XML documents in compliance with the LC/SCF in addition to implementing the client to service broker Web service interface. More elaborate clients, such as that of the Microelectronics WebLab (Fig. 3), employ graphical mechanisms for representing lab resources and robust tools for graphing and exporting experiment data. The WebLab client has also been developed in a modular way to encourage the reuse in other lab clients of components such as the graphing engine and the service broker communication module. Thus far, Java has been the preferred client development environment due to its ubiquity as an execution environment and its portability across computer platforms/operating systems. Other client technologies that have been tested include Windows Forms and PHP/Ajax/JavaScript.

At the other end of the system, the lab server receives experiment requests from service brokers, operates lab instrumentation in order to perform experiments, and delivers results to the originating service brokers. The lab server must implement the service broker to lab server Web service interface, be able to interpret and produce documents in compliance with the LC/SCF, and be able to translate experiment parameters from the client to lab instrument commands.

In the batched experiment scenario, multiple users can submit experiments for execution simultaneously, and a given lab server can receive experiments from an arbitrary number of service brokers. Most lab servers, therefore, require an experiment queuing system of some sort. The WebLab lab server implements this queue using a standard relational database. This persistent data store forms the core of the lab server application.

The WebLab lab server operates as two complementary processes: a Web interface module and the experiment execution engine. The persistent data store, which runs as a third (database) process, connects these two modules (Fig. 4). The Web interface implements the Web services that the service broker invokes to submit experiments. It also includes experiment validation



#106507 (1/17/08 1:36:12 PM) NMOS output (3 terminal)
Temperature: 298.6 K

Y1 Axis: 100.0 mA; 0.000 A; Variables: ID; Scale: Linear; ☐ autoscale

Y2 Axis: 200.0 mS; 0.000 S; Variables: GD; Scale: Linear; ☐ autoscale

X Axis Var: VD Scale: Linear ☑ autoscale

Fig. 3. ***Screen shot of the microelectronics WebLab Java client showing the experiment description panel and the graphing module. In this screen shot, a metal–oxide field-effect transistor is being characterized.***

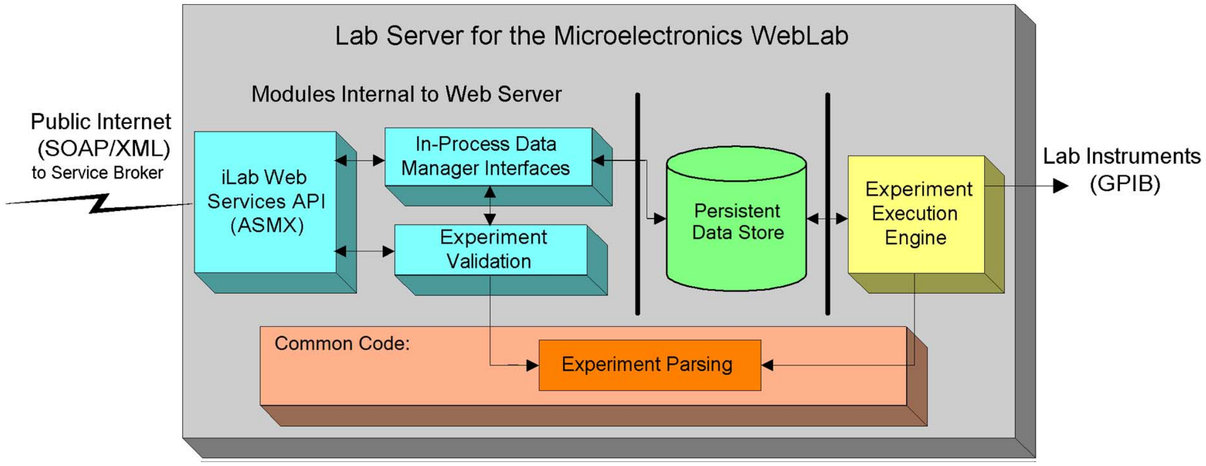


Fig. 4. ***The structure of the batched lab server.***

methods and a lab administration interface. When a service broker submits an experiment, the Web interface module validates that request, places it into the experiment queue, and monitors its progress.

The experiment execution engine monitors the queue and retrieves new experiment submissions in order to execute them on the lab hardware. Execution simply requires translating a request's XML-based encoding scheme into instrument-specific execution commands. When the experiment completes, the experiment execution engine updates the data store with the results. This triggers the Web interface module to notify the appropriate service broker that those results are ready to be retrieved.

The persistent data store acts as a message board of sorts. By communicating through the data store, the other lab server processes are able to operate in a coordinated fashion while remaining independent of each other. Shared library classes implement common functionality, such as the methods for parsing messages encoded with the lab-specific LC/SCF. Otherwise, there is minimal overlap between these independent processes. Not only does this provide a clean division of functionality across the lab server but it also results in a system that provides better experiment throughput and more reliability than previous, more monolithic designs. Much of the code in these three modules can be reused in the construction of new online labs.

Since students can prepare experiments destined for the same lab server in parallel on separate clients, batched labs scale extremely well to large numbers of users. If two users submit their experiments at the same time, the lab server will queue them and execute them in quick succession. The online version of the Microelectronics WebLab can increase the throughput of the expensive Agilent semiconductor parameter analyzer by up to two orders of magnitude compared with the traditional approach of keying in the characterization parameters on the front panel.

The batched architecture also provides very robust performance in situations of poor network connectivity between the client and the lab server because the client need not remain online while the experiment executes. The service broker will poll the lab server to retrieve the results once the experiment has completed, and the client can reconnect later to display them. In the case of the Microelectronics WebLab, this has improved reliability for users at several universities in Africa who were experiencing significant network and infrastructure instability.

D. The ISA Interactive Middleware

1) The Network Topology of the Interactive Architecture: In the iLab batched architecture, all communication between the client and lab server passes through the batched service broker (Fig. 2). Should the interactive service broker (ISB) play the same central role in the interactive architecture? Routing all communication through the ISB would allow it to save an authoritative log of the user's control of the experiment and the corresponding results. It would also simplify authentication and authorization. On the other hand, it would increase network latency between the interactive client and the lab server. Every control message from the client to the lab server and every status or result message from the lab server to the client would require two network hops instead of one.

The lab development community strongly urged the case for allowing direct communication between client and lab for a second reason. Interposing the ISB would restrict lab and client developers to using an iLab defined protocol for passing control and result information, as in the batched architecture. Many labs today are computer controlled even if remote access is not a requirement. Lab developers often create a virtual interface to the lab that runs on a second system using virtual instrumentation packages like LabVIEW. This (user) client and (lab) server system is often

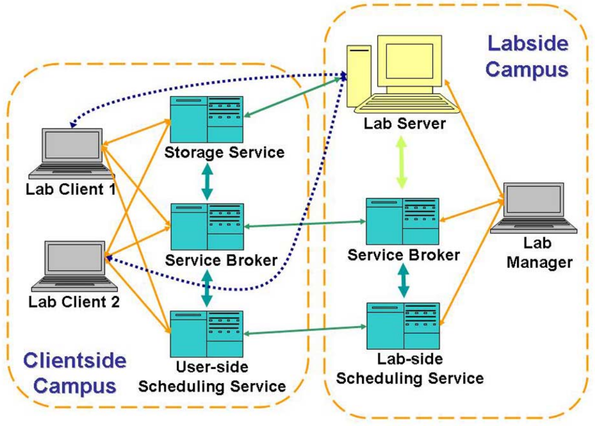


Fig. 5. ***Topology of the iLab architecture for interactive experiments.***

built before the decision is made to move the system to the iLab architecture. Allowing direct communication between the user client and lab server gives developers the freedom to choose their own communication protocol and to use third-party packages like LabVIEW and MATLAB in their development. Thus the iLab team decided to allow direct communication even though it was going to introduce added complexity to the interactive design (Fig. 5).

2) The Experiment Storage Service: If the interactive client and lab server can communicate directly, the ISB cannot be responsible for creating the definitive record of the experiment, as in the batched architecture. In fact, all three processes—the ISB, the client, and the lab server—may need to store information to record the complete experiment. This suggests that an independent experiment storage service (ESS) be exposed as a Web service to handle the potentially high-bandwidth traffic from the client and lab server during experiment execution. The ESS stores both XML and binary data from a particular experiment but does not store the corresponding administrative information, e.g., the owner of the experiment log, the class the experiment was executed for, etc. This administrative information is managed by the ISB. Thus, any user request to access an experiment log on the ESS must start with locating the experiment on the ISB since only the ISB can relate users to their experiments. The ISB then authorizes the user's client or analysis program to retrieve experiment data from the ESS.

3) The Scheduling Services: The interactive architecture permits students to observe the progress of an experiment and to change the experiment's course in real time. Such labs typically require more time to execute than batched experiments because they proceed in human, not machine, time. A typical interactive experiment needs 20 minutes to several hours to execute. Since users control the lab equipment while their experiment executes, they usually require exclusive access to the apparatus. Hence users of interactive experiments usually request a scheduling application that will allow them to sign up in advance for time on a particular piece of lab equipment. Access to this scheduling application must be authorized by the ISB, since only the ISB can authenticate a user and vouch for his or her right to schedule a reservation. The scheduling application should also notify users if their reservation must be cancelled or changed. Finally, certain labs have operating requirements that require actions either before or after the execution of an experiment. For instance, a chemical diffusion lab employing a dye solution may require that the diffusion tanks be flushed at the end of the experiment. The scheduling application must allocate time for these actions while reserving experiment sessions.

Scheduling can be looked at from two perspectives. From the lab provider's perspective, the scheduling application coordinates reservations to use the lab from multiple campuses. The scheduling server is also the process that holds the information required to "wake up" a lab server to perform required actions before a scheduled experiment. On the other hand, the lab provider generally does not want to be aware of the details of a particular user's reservation. If the lab server must be taken down for maintenance, the lab provider would simply like to notify the scheduling application of the down time and have the scheduling application take care of informing the affected users and rescheduling their work.

From a teacher's and a student's perspective, the scheduling application must act as their agent in scheduling time on lab servers. The application must accept authorizations to schedule from the users' ISB and must record reservations in a way that can be associated with individual users. A student should be able to change or cancel a previously made reservation. If lab maintenance forces the cancellation of reservations, the scheduling application must take the responsibility for informing the user. Teachers may want to stipulate policies that govern how their students may make reservations. For instance, a teacher may decide that students can only sign up for two hours of lab access per week, with no single reservation lasting more than one hour. Different teachers using the same lab may want to set different policies for their students.

Given the different requirements from the lab-side and the student-side perspectives, where should the scheduling application be located? The need to coordinate reservations from multiple campuses for a single lab server argues that there should be a single scheduling application acting as gate keeper for a lab server. But the requirement to accommodate the different policies of individual teachers suggests the need for multiple scheduling applications, typically one on each student campus as in the case of batched service brokers. We have decided that the two perspectives require two related scheduling applications: a lab-side scheduling server (LSS) and a user-side scheduling server (USS).

The two scheduling applications communicate using a very simple and restricted Web service protocol. All the intelligence and complexity is housed within the two applications. Their initial implementations support only a simple set of scheduling policies: e.g., on the USS first come, first served, limited by a maximum reservation allowance and a maximum reservation length. Decoupling the LSS and USS allows the development of each to proceed independently. For example, a university that wants to implement an innovative user scheduling policy can do so without needing to modify the scheduling policy for the lab server whose LSS may be located at another university and controlled by different staff (Fig. 5).

4) Authentication and Authorization: To begin either an administrative or experimental session with the iLab interactive architecture, the user must be authenticated by the ISB. The reference implementation supplies a simple user name and password scheme carried out using a standard browser-based Web application. The architecture permits other authentication mechanisms, e.g., authentication by certificate, to be added to the implementation.

Once the ISB knows the identity of the user, it supplies authorizations for actions that the user wants to perform on other distributed applications and servers. Two examples follow.

1) After authentication, the user indicates that she wishes to schedule a future lab session and chooses one of the labs to which she has access. Then the ISB redirects her to the Web application of the USS that handles the reservations for that lab. The redirection must be accompanied by credentials sufficient to identify the user and to convince the USS to allow her to schedule a future experiment session.
2) When the time has come for the student to execute the experiment, the ISB must launch the client with credentials that the lab server will recognize. The client will usually contact the lab server directly, and the lab server should only accept the connection if it trusts the credentials originally furnished by the ISB. These credentials include the period the user has reserved and the group (or class) for which the experiment is being executed. Different groups may be allocated different levels of access. A graduate class may be able to perform more sophisticated functions than an introductory class.

The second example introduces an additional requirement because the lab server will probably need to use the same credentials to invoke services on behalf of the user. When the lab server needs to store experiment data, it must contact the user's ESS and present the forwarded credentials that will allow the ESS to recognize who owns the data that is being stored.

The interactive architecture currently employs an iLab-specific credential mechanism known as *general ticketing* [25]. In general ticketing, a user's browser or client never actually holds the credentials themselves but only a receipt for the credentials called a *ticket coupon*. A lab server or other service provider uses the coupon to retrieve the actual credentials from the ISB. This prevents a user from forging credentials. The project is currently conducting a review of Web application and Web service security mechanisms including Shibboleth, WS-Security, and SAML2 to determine the best strategy for converting iLab authentication and credential management to a cross-platform and standards-based infrastructure.

5) The Architecture of Interactive Lab Servers: The interactive lab server (ILS) is responsible for processing experiment execution requests. Once a request is validated, the ILS initializes the experiment, feeds data back to the user interface in the client, stores experimental data on the ESS, and closes down the experiment after the reservation has expired. The ILS is not responsible for scheduling but may need to respond to alerts from the LSS. The sample implementation of the ILS included with the standard software distribution has been designed with an abstraction layer that segregates generic modules from lab dependent code. These generic modules (authorization, experiment validation, an experiment life-cycle manager, and a generic ESS interface) are independent of the actual experiment's technology.

E. The ISA-LabVIEW Interface

One goal of the iLabs project is to provide a rapid conversion path from an existing standalone lab to an Internet-accessible one. The wide acceptance of National Instruments LabVIEW convinced us to release a LabVIEW Integrated Interactive Lab Server (LVILS). Built upon the ILS generic classes, the LVILS provides interfaces between the .NET 2.0 Web Service application of the ILS and LabVIEW processes.

The lab experiment's front panel is displayed using the LabVIEW Remote Panel Server, either within a .NET page or via the LabVIEW Web Server. Simple support for reading and writing data to the ESS is provided by a `DataSocket` implementation of the generic class `LabDataSource`. A .NET interface built around the LabVIEW VIServer ActiveX control provides management of the LabVIEW process and an individual experiment's virtual instruments (VIs). VIs are the components that define LabVIEW programs. The VI server restricts certain remote operations such as the ability to disconnect a user due to security requirements. A collection of iLab-supplied generic VIs provides a way to implement such restricted operations under safe iLab control. Thus if a user overstays his previously scheduled reservation to use a lab, the LVILS can terminate his session.

Once a developer has implemented a standalone version of a LabVIEW experiment, the process of converting that application to run under the iLab interactive architecture usually requires only two standard steps that can be accomplished in hours.

1) The developer must use instances of the `DataSocket` class to write data to the ESS from the lab server using an experiment-specific XML record format.
2) The developer should develop experiment-specific VIs to handle lab reset and shutdown.

V. THE PEDAGOGY OF ONLINE LABORATORIES

The iLab project does not consider iLabs to be a substitute for hands-on experience in a physical lab. The project is trying to determine best practices along a number of dimensions:

- What experiments are best suited to be presented as an iLab?
- Is there an appropriate integration of online and hands-on laboratories that is optimum for a given lab experience in a given subject?
- What principles should guide the design of the client interface presented to the students?
- What pedagogical materials need to be given to the students before they use an iLab to best engage their interest and optimize their learning as they work their way through the lab.
- How can course staff provide online support to students who are executing the lab remotely at random times? (In the case of assigned experiments on the Microelectronics WebLab, staff has noted that student usage routinely peaks after midnight on the night before an assignment is due (Fig. 6).

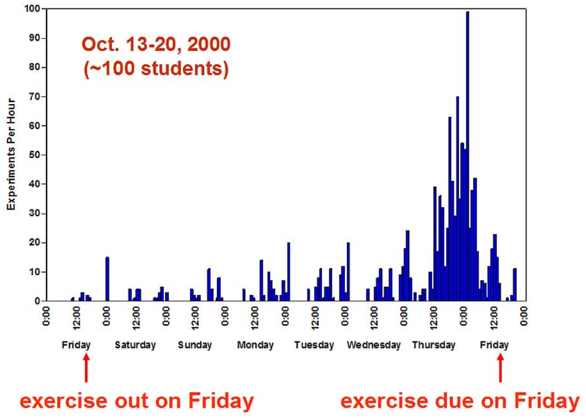


Fig. 6. ***Experiment executions per hour in an assignment to a junior level class with about 100 students in October 2000. The assignment went out on a Friday afternoon and was due the following Friday afternoon. The peak of activity took place in the early morning hours of the day when the assignment was due.***

The project has carried out both informal and formal evaluations of the use of iLabs in undergraduate courses. Two case studies have thrown particular light on the educational role of online labs.

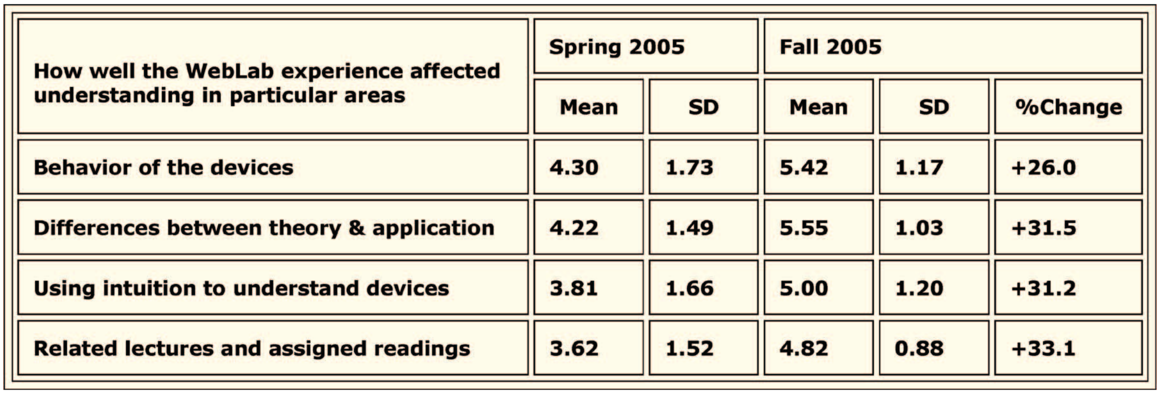
The MIT Microelectronics WebLab Case Study: By the fall of 2004, the Microelectronics WebLab had been used at MIT in graduate and undergraduate subjects in electrical engineering for several years. Yet, a formal study of the effectiveness of this lab still had to be carried out. In the fall of 2004 and the spring and fall of 2005, the MIT Teaching and Learning Laboratory launched an evaluation of the use of this lab in 6.012 "Microelectronics Devices and Circuits," an elective junior-level (third-year) subject in electrical engineering at MIT [26]–[28].

The evaluation consisted of quantitative surveys and in-depth qualitative interviews with students, instructors, and teaching assistants in the course. The fall 2004 edition of 6.012 was used by the evaluators to carry out a pilot study that would help to develop the methodology and identify the issues to be investigated in the two subsequent semesters. Surveys and interviews carried out in the spring of 2005 identified a number of areas for improvement. In response to these findings, several changes were made to the use of the lab in the subject for the fall of 2005 when it was evaluated again.

In 6.012, through the Microelectronics WebLab, students measure the current–voltage characteristics of microelectronics devices (such as diodes and transistors), obtain device parameters, develop device models, compare theoretical models against experimental data, and comment on discrepancies. These assignments involve experimental work followed by data manipulation and the development of computer programs to model device behavior. In the spring of 2005, students were asked to carry out two extensive device characterization projects requiring several hours of work each. In response to the evaluation results that were obtained, these assignments were broken into smaller portions that were sprinkled through the regularly scheduled homework for the following fall semester. Staff also enlarged and rewrote the system documentation, introduced on-line tutorials, and corrected several bugs and other incorrect information.

The results of this study were very encouraging. In each of the semesters examined, surveys and interviews of student perceptions yielded evidence of improvement in teaching and learning with WebLab. Students shared their enthusiasm for using the system's clear and coherent graphic interface, reporting that remote control of lab equipment brought a welcome gain in time-efficiency and did not interfere with learning. Indeed, most observed that using WebLab enhanced conceptual learning, stimulated higher order thinking, and reinforced individual styles of learning in multiple ways. The program allowed students to control their own learning processes while enabling faculty to maintain factual rigor and coherence throughout

Table 1 Sample of Student Survey Results in the Evaluation of the Microelectronics WebLab in a Junior-Level Subject in Electrical Engineering at MIT. The Results are Ranked on a Seven-Point Likert Scale, Ranging From "1" (Poorly) to "7" (Extremely Well) [26]–[28]



How well the WebLab experience affected understanding in particular areas	Spring 2005		Fall 2005		
	Mean	SD	Mean	SD	%Change
Behavior of the devices	4.30	1.73	5.42	1.17	+26.0
Differences between theory & application	4.22	1.49	5.55	1.03	+31.5
Using intuition to understand devices	3.81	1.66	5.00	1.20	+31.2
Related lectures and assigned readings	3.62	1.52	4.82	0.88	+33.1

the course. A sample of quantitative results is shown in Table 1 [27].

The conclusion from this study was that the Microelectronics WebLab is a successful Internet-based resource that offers students the freedom to choose from a variety of learning strategies and to apply them in effective combinations suited to individual needs. Because of these strengths, this approach holds great benefits for the teaching of many empirical disciplines.

The University of Queensland Inverted Pendulum Case Study: The University of Queensland's adoption of the iLab batched architecture for the delivery of an existing lab provides the best case study to date of the contrast between a traditional and an iLab version of the same experiment.

The inverted pendulum is a classic control theory experiment wherein the student attempts to balance a pendulum with the weighted arm pointing upright towards the ceiling rather than hanging towards the floor. There are several implementations of this concept that exist, but in this example it consists of an actuated arm attached to a freewheeling pendulum arm. The pendulum is balanced by moving the actuated control arm back and forth, swinging the pendulum up and then catching and balancing it in the upright position (Fig. 7) [29].

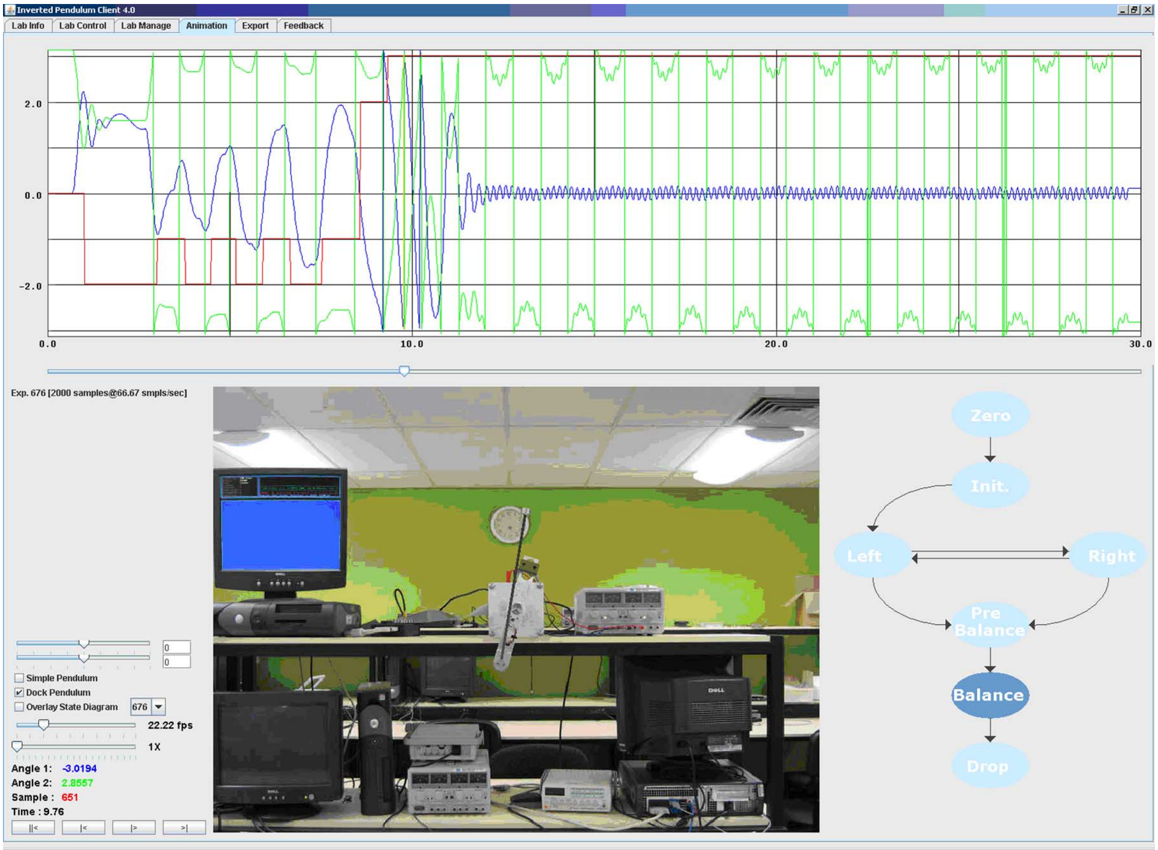
The University of Queensland has used a traditional form of this experiment in a course of 80 students, with five inverted pendulums for students to share during the three or so hours of allotted lab time each week. Students initially felt constrained by their limited access to the lab equipment. Converting the inverted pendulum to an online lab using the iLab architecture significantly increased students' access to the experiment because they could now use it when the physical lab was closed. This, in turn, led the students to spend more time using the equipment on their own initiative, and consequently there was a dramatic increase in the proportion of students who successfully balanced the pendulum during the portion of the course devoted to the lab (from 4% to 73%).

Very little of the students' experience of the lab was lost through the abstraction to the virtual interface (Fig. 7). In the traditional version of the lab, the students would write a Simulink control model, upload it into the computer driving the actuated arm, and then observe the behavior of the system in real time. In the iLab version of the experiment, the students submit the same control model as a batched experiment description. The model executes until it reaches the exit condition, and the behavior of the pendulum is recorded so that it can be replayed through the iLab batched client.

In fact, through the remote client interface, students had substantially more insight into the results of their experiments than they had through physically interacting with the equipment in the lab. The remote interface allowed them to watch the balancing process multiple times in slow motion, if desired, and observe the internal state changes their control model underwent during a particular experiment run. It also allowed students to compare the behavior of the pendulum during multiple runs of the experiment using different control models. Even during the weekly allotted lab times, when students had physical access to the equipment, most still chose to use the iLab version of the experiment.

Course staff found that implementation under the iLab architecture also gave course coordinators the ability to better monitor the students' usage of the equipment. Safeguards were put in place to terminate experiments that were damaging the equipment (e.g., by violent shaking), resulting in less damage to the limited number of experiment setups. It also allowed for the detection of plagiarism since the system kept extensive logs of who ran what experiments when and the exact control models they used.

Evaluation confirmed that converting the inverted pendulum to an online lab with a more informative interface led to an improvement under every category of concern in which a course instructor would be interested. Students learned more, did better, and were happier with the experiment. At the same time, the staff had greater



Exp. 676 [2000 samples@66.67 smpls/sec]

0
0

☐ Simple Pendulum
☑ Dock Pendulum
☐ Overlay State Diagram 676

22.22 fps
1X

Angle 1: -3.0194
Angle 2: 2.8557
Sample : 651
Time : 9.76

Fig. 7. ***The execution panel of the University of Queensland inverted pendulum experiment showing the pendulum visualization, execution trace, and animated state diagram.***

confidence in the integrity of the students' work and less worry regarding the maintenance of equipment in the face of heavy student use.

VI. THE GROWTH OF AN iLAB COMMUNITY

Since its inception, the iLab project has emphasized the sharing of labs. Unlike conventional labs that every institution must own and maintain, iLabs can be shared worldwide around the clock. iLab experiments have been used in courses by 18 universities in Europe, Australia, Africa, Asia, and the United States. Over the past four and a half years, iLabs at MIT alone have performed more than 73 000 experiments for 2400 MIT students and roughly 3300 other worldwide users. MIT faculty are developing iLabs in several fields including physics and electrical, chemical, and nuclear engineering.

A growing number of institutions beyond MIT have committed to putting labs online using the iLab architecture. iLab-China is an informal group of Chinese universities, led by Dalian University of Technology, that are developing new iLabs with the support of MIT students under the MIT International Science and Technology program. Dalian is currently adapting an existing lab that controls an air conditioning system to use LabVIEW and the iLab middleware for remote access.

The most dramatic adoption of the iLab approach, however, has occurred in Australia. iLab-Australia is a growing consortium of Australian universities led by the University of Queensland and includes The Royal Melbourne Institute of Technology (RMIT) and the University of Technology, Sydney. They have already put three iLabs online and have nearly a dozen others in various stages of development or planning across a wide variety of engineering disciplines. In fact, the success of the iLab architecture in Australia has far outstripped its spread in the United States. Not surprisingly, iLabs have also attracted great attention in developing countries. The patterns of adoption there have had to meet very different challenges than those in the United States, Australia, and Europe.

A. Slow iLab Adoption in the United States

The effort to disseminate and support the adoption of the iLab project in the United States has had admittedly weak results. Project members communicated the availability of this technology to universities and colleges around the United States through a number of channels: professional meetings (e.g., the Microsoft Research

Faculty Summit,[7] the EDUCAUSE Annual Meeting, the Sloan Asynchronous Learning Network, the International Conference on Engineering Education), direct outreach visits to selected universities, iLab workshops, and the iCampus Web site,[8] among other strategies. These activities complemented faculty directly engaged in iLabs who leveraged their professional networks to carry the message of iLabs to their colleagues.

The iCampus project also suggested and sponsored an open portal to selected remote laboratories at MIT.[9] Through this portal, students, educators, and self-learners can gain unrestricted access to some of MIT's iLabs. Currently, three iLabs are available: the Microelectronics WebLab, the Dynamic Signal Analyzer [30], and the ELVIS circuit lab. This portal has also allowed colleagues to sample the use of several iLabs before committing to install a service broker or develop a new iLab.

We have come to believe, however, that there are factors in American higher education that tend to work against the adoption of a cross-institution technology such as iLabs. Australia, in contrast, has provided an environment in which iLabs have flourished, in large part because the Australian universities are confronting different challenges than their U.S. partners. We believe this contrast in adoption has general implications for interinstitutional collaboration on educational software.

The reaction to iLabs within the United States has tended to fall into two categories: that of potential lab providers and that of potential lab consumers. American universities that already possessed labs that could be converted to online usage have generally failed to see value commensurate to cost in increasing access for their own students. And if there is little perceived need to increase access for a university's own students, it becomes nearly impossible to make a case for investing in technology and staff to share such facilities with other universities unless outside funding is available.

"Consumer" institutions that have greater need for lab access have repeatedly expressed the wish for a catalog of openly available, "free" experiments. iLabs currently provide a mechanism for sharing experiments, but the project has never sought funding to implement a broad repository of laboratory exercises geared toward the standard curricula at consumer institutions.

The recent collaboration between MIT and MATEC,[10] a consortium of U.S. community colleges with strong electronics programs, could be the sole U.S. example hitherto where an iLab producer has partnered with consumers to produce a highly leveraged lab and set of curricula. MATEC approached MIT to lead the development of a new electronics iLab as part of the New Systems View of Electronics 2010 Project sponsored by the National Science Foundation. MATEC will provide the laboratory exercises and curricula for the project. In effect, both groups have become producers of complementary materials that can be shared in the iLabs environment.

The summative iCampus assessment report [31] has pointed out another factor limiting adoption in the United States. The entrepreneurial nature of American higher education highly prizes innovation. Structures have evolved to support the research enterprise and the faculty engaged in it. The iLab project, however, has now reached its dissemination and adoption phase. That is, the work associated with bringing a new iLab online by a "producer" institution involves adapting an existing laboratory, or acquiring the equipment for a well-understood laboratory and implementing it on top of the iLabs software infrastructure. A "consumer" institution develops laboratory exercises that match its curriculum. Institutional structures to support faculty converting existing experiments to a new software architecture like iLabs are largely absent. The participating faculty member must be software integrator, curriculum developer, instructor, support staff, advocate, and outcomes assessor. But these activities seldom contribute to a tenure file or secure large grants in the faculty member's main discipline.

B. Rapid Adoption in Australia

One of the authors (P. Long, in his role as director of the MIT iCampus Learning Outreach Project) introduced Australia to the concept of iLabs through a talk presented at the University of Queensland (UQ) in late 2004. This talk occurred just as UQ and other Australian universities were trying to address a particular set of new challenges.

UQ was struggling with finding the funding and space to expand laboratory facilities across multiple campuses. Previously, laboratories for large classes (100–300 students) were built with 25 duplicate workstations at which pairs of students worked. Lab time was scheduled so tightly that students had only a single opportunity to complete experiments. iLabs offered the promise of using a smaller existing space to install fewer sets of experiment equipment and making it available around the clock to more students through online sharing.

UQ was also eager to give local high school students access to some of the laboratories on the campus. Such lab access was fraught with legal problems involving workplace safety that seemed to be intractable, but iLabs offered the ability to support these additional students with little to no additional cost or risk. UQ administrators also realized that if faculty could present the online experiments effectively, it might well encourage students to enroll at the university.

With the political groundwork laid, work began on the first iLabs experiment at UQ. Development fell to another of us (J. Carpenter), still an undergraduate, as a final year

---

[7] For example, http://www.research.microsoft.com/workshops/fs2007/.

[8] http://icampus.mit.edu/outreach/.

[9] http://openilabs.mit.edu.

[10] http://www.matec.org.

project. Communicating freely with the iLabs team at MIT, Carpenter spent his first semester learning the architecture and building a test system. During his final semester he converted the existing hands-on inverted pendulum experiment discussed above to run as an iLab. A class of engineering students then successfully used the new iLab during the final weeks of that same semester.

Technical competence alone would not have ensured the success of iLabs at UQ. Another of the authors (M. Schulz), a member of the UQ engineering faculty, became the local champion for the iLab project within UQ and around Australia. He roused the interest of his colleagues within electrical and computer engineering to adapt experiments to run under the new architecture. As the iLab concept continues to spread around Australia, the need to find a dedicated early adopter at an institution is still a major precondition for success.

The continued involvement of the staff from MIT has also remained important. There have been visits to Australian universities by MIT staff each year for an annual meeting. On one occasion, a large portion of the MIT iLab team visited to help present a developer workshop.

Meanwhile, the senior leadership team at UQ had the desire to see the university take on an international role in innovative teaching and learning practices, and they believed iLabs, as well as other projects in the iCampus portfolio, provided a means to pursue this goal. They have provided liberal financial backing not only to foster teaching and learning development at UQ but also to assist other Australian universities to adopt and adapt iCampus tools. UQ has funded Schulz as the director of the Australian iCampus Dissemination to travel to Australian and Asian universities to promote iLabs and iCampus. This joint support from both UQ and from MIT has helped to convince other Australian universities that iLabs are worthy of their institutional investment.

RMIT joined the iLab effort in 2004, when the UQ deputy vice chancellor, responsible for determining the strategic directions for teaching and learning at UQ, left to become vice chancellor of RMIT. RMIT now has their first iLab experiment (a synchronous power generation lab) up and running. Regular iCampus/iLab seminars at Australian universities and an annual Pan-Australian iCampus Workshop have been held since 2005. The growing expertise of Australian universities with iLabs has contributed to the important perception that this is not solely an MIT technology.

Looking forward, UQ has recently purchased a campus-wide license for LabVIEW to control laboratory equipment and experiments. The intention is to leverage the integration of LabVIEW with the iLab interactive architecture to create many new (teaching and research) experiments available to a far wider audience. UQ has also appointed a full-time technical support staff member dedicated exclusively to iLabs and LabVIEW.

C. iLabs in Developing Countries

Sponsored by the Carnegie Foundation of New York, MIT has formed a partnership (iLab-Africa) with three sub-Saharan universities: Obafemi Awolowo University in Ile-Ife, Nigeria; Makerere University in Kampala, Uganda; and the University of Dar es Salaam in Tanzania. The goals of the iLab-Africa project are to explore and exploit the promise of online laboratories to enhance science and engineering education among universities in Sub-Saharan Africa.

iLabs offer the potential of enriching education around the world by bringing educationally meaningful laboratory experiences to students wherever an Internet connection is available. This "universality" of iLabs is seriously challenged in locations where access to the global Internet is limited by narrow bandwidths and high connection costs, or where students have restricted access to computers. Such conditions are pervasive in the developing world and are particularly dire in sub-Saharan Africa. Realizing the potential of iLabs in these environments requires more than just providing free access to existing iLabs around the world to African students. It requires a close collaboration among educators to develop and share educational content. It also demands technology transfer and adaptation as well as personnel development to promote the creation of new iLabs designed to address unique curricular goals and constraints.

A feasibility study that we carried out in 2003–2004 showed that iLabs hold the potential to have a comparatively greater impact in academic institutions in the developing world than in the developed world. This is because of the paucity of hands-on laboratory experiences available to science and engineering students in developing countries. Against this, iLabs can provide access for teachers and students to state-of-the-art tools, devices, and systems. The insertion of iLabs into the curriculum would also bring to the fore the power of the computer as a versatile engineering tool: students will be exposed to data acquisition, analysis, interpretation, and model development. Our feasibility study also revealed serious challenges. Some of them are structural such as narrow bandwidth, restricted access to networked computers, and very tight budgets. Some of them are cultural, such as insufficient student exposure to computers and a culture of limited institutional support for personal teaching and learning tools.

The iLab-Africa project has made considerable progress in attacking these challenges. On the technical side, we have identified approaches that mitigate the bandwidth bottleneck. Bandwidth is a problem not only in its limited quantity but also in its poor quality. The quantity is limited by the high cost of satellite links. Even in countries where there are fiber-optic landings from submarine cable systems, national networks do not penetrate deep enough into the country. Bandwidth "quality" is also an issue since campus networks experience many glitches and electrical power is unstable.

The scarcity of high-quality bandwidth impacts the use of iLabs in several ways. When downloading a client application directly from a server across the world, applet corruption was a common problem due to the long downloads and the increased likelihood of suffering a network glitch during download. System responsiveness is also an issue. Network applications are much less responsive and this impacts student engagement and limits the effectiveness of the educational experience. These issues are addressed to a great extent by the iLab architecture. Installing a service broker inside the African campus and downloading the client from this location greatly reduces download times and mitigates the likelihood of client corruption. In experiments performed at OAU, we found that downloading and initializing the Microelectronics WebLab client from an MIT server took on average 79 s, while if it was downloaded from a local service broker installed in the OAU intranet, it took only 22 s.

Restricted access to networked computers was also found to be a difficulty. In general, university computer clusters seem unable to fulfill student demand: computer clusters are few and small, and the hours are restricted. In addition, student ownership of personal computers is relatively rare. In order to carry out their assignments, students are forced to use computers in computer cafes at a typical cost of about $1/h. This comes with limitations. Many computers in Internet cafes do not have an up-to-date Java plug-in required by typical Java clients written for U.S. universities. In response to this, we took advantage of the fact that the iLab architecture supports multiple clients for a given lab and developed a "nimble" client specially designed for developing countries. This thin client is very compact, does not require the use of a plug-in, and employs fewer graphical elements. We found that this client downloaded to OAU in 63 s from MIT and 17 s from a local service broker.

In the first two years of operation of the iLab-Africa project, nearly 700 African students have used MIT's iLab experiments in their courses. iLab development groups have been created at each of the African universities and have begun to develop their own experiments. OAU has successfully launched its first iLab experiment, a platform for taking measurements in operational amplifier-based circuits in which students can configure a circuit around an op-amp and measure the transfer characteristics of the entire system. The development of the second experiment, a digital logic lab, has just finished, and it will be tested in a class in early 2008.

D. The Future of iLabs: The iLab Consortium

MIT's growing partnership with other institutions has led us to realize that the goals of the iLab project must go beyond just sharing access to laboratory equipment using a common infrastructure. They must also include the creation of a scalable and sustainable online community where faculty, students, and researchers, from around the world, come together to share and collaborate on iLab-based curricular materials and teaching experiences.

MIT is currently developing an iLab community site to host pedagogical materials including lab and problem set descriptions as well as evaluation reports on iLab use.[11] We hope the site will form the nucleus for a community brought together by their common interests and provide a framework for making high-quality reusable curricula for educators. Through this site, the iLab project expects to support the evolution of a community of practice to enhance science, technology, engineering, and math education. The site is intended not just as a platform for sharing information and expertise but also as a forum through which the various concerned communities can debate the evolution of this technology.

Since the very first release of the iLab software, MIT has made documentation, source code, and sample lab-server code available to all on an open-source basis using a variant of the OpenBSD license.[12] Until recently, MIT had undertaken all development of the ISA middleware, but UQ has started to make valuable software contributions. The iLab project welcomes partners to join in the future design and implementation of the ISA across multiple machine and OS architectures.

The ultimate organizational structure for the iLab project should probably be a consortium of academic and commercial partners committed to the growth of the technology and the associated educational resources. This implies that the membership of the consortium should include both iLab providers and consumers. Commercial partners may be interested in interfacing their hardware and/or software technologies to the ISA middleware. The iLab OpenBSD license also allows the possibility of a commercial version of the iLab middleware with the advantages of bundled installation packages with commercial grade documentation and support. One of the challenges such a consortium must face will be to balance the respective contributions and goals of all the members in such a way as to foster the growth of a broad economy of online labs. Such an economy should include market mechanisms for the efficient trading of spare lab time around the world.

The pace of iLab adoption has increased dramatically over the past year. Discussions with other potential iLab partners are in progress. We feel that, while the iLab project may not have reached maturity, it has certainly entered a robust adolescence that transcends its beginnings at MIT.

VII. CONCLUSION

The iLab Shared Architecture provides a flexible software infrastructure for the implementation of Internet-accessible labs. The underlying concept of online labs, the

---

[11] http://confab.mit.edu/confluence/display/ILAB2/Home.

[12] http://icampus.mit.edu/iLabs/Architecture/Downloads/default.aspx.

middleware technology to support them, and the pedagogical expertise to guide their use in teaching has evolved over a decade's research and development at MIT. The fruits of this work are now freely available, and MIT has been joined recently by new partners who are rapidly broadening the scope of the project to meet needs and challenges that were not envisioned at the project's start. We expect to form an iLab Consortium in the near future to broaden the political and technical leadership of the project as well as to foster a greater exchange of lab resources and curricular materials between partners located across the globe. ■

Acknowledgment

The authors wish to thank present and former colleagues at MIT and Microsoft whose experience and insights have helped guide the development of the iLab Shared Architecture: C. Colton, G. Rutledge, K. Amaratunga, H. Abelson, J. Maro, P. Mitros, A. Jiwaji, S. Gikandi, A. Haldar, B. Cukalovic, D. Mitchell, and P. Oka. M. Westlund has provided crucial administrative support to the project for years and valuable editorial assistance in the preparation of this article. Our colleagues at OAU, particularly K. Kehinde, E. Ajayi, O. Osasona, K. Ayodele, O. Ilori, P. Jonah, and O. Akinwunmi, have dramatically broadened our concept of the project's goals and requirements as well as validating our hope that iLabs can become a truly global resource. A. Johnson of the Carnegie Corporation of New York has always shared her broad knowledge of Africa with us generously. R. Almgren and A. Watchorn of National Instruments have consistently challenged us to think beyond the bounds of MIT's sometimes blinkered outlook. The Dean of Engineering at MIT during iCampus, T. Magnanti, nurtured this project through the development of the iLab Shared Architecture.

REFERENCES

[1] J. Ma and J. V. Nickerson, "Hands-on, simulated, and remote laboratories: A comparative literative review," *ACM Comput. Surv.*, vol. 38, no. 3, Sep. 2006.

[2] L. Gomes and J. García-Zubía, Eds., *Adv. Remote Lab. e-Learn. Exper.* Bilbao, Spain: Univ. of Deusto, 2007.

[3] J. A. del Alamo, V. Chang, L. Brooks, C. McClean, J. Hardison, G. Mishuris, and L. Hui, "MIT microelectronics WebLab," in *Lab on the Web*, T. A. Fjeldly and M. S. Shur, Eds. Hoboken, NJ: IEEE Press/Wiley, 2003, ch. 2, pp. 49–87.

[4] J. A. del Alamo, L. Brooks, C. McLean, J. Hardison, G. Mishuris, V. Chang, and L. Hui, "The MIT microelectronics WebLab: A Web-enabled remote laboratory for microelectronics device characterization," in *2002 World Congr. Netw. Learn. Global Environ.*, Berlin, Germany, May 2002.

[5] J. A. del Alamo, J. Hardison, G. Mishuris, L. Brooks, C. McLean, V. Chang, and L. Hui, "Educational experiments with an online microelectronics characterization laboratory," in *Proc. Int. Conf. Eng. Educ. 2002*, Manchester, U.K., Aug. 2002.

[6] J. A. del Alamo, V. Chang, J. Hardison, D. Zych, and L. Hui, "An online microelectronics device characterization laboratory with a circuit-like user interface," in *Proc. Int. Conf. Eng. Educ. 2002*, Valencia, Spain, Jul. 2003.

[7] K. Jeppson, P. Lundgren, J. A. del Alamo, J. L. Hardison, and D. Zych, "Sharing online laboratories and their components—A new learning experience," in *Proc. 5th Eur. Workshop Microelectron. Educ.*, Lausanne, Switzerland, Apr. 15–16, 2004.

[8] C. K. Colton, M. Q. Knight, R.-A. Khan, and R. West, "A web-accessible heat exchanger experiment," in *Proc. World Innov. Eng. Educ. Res. (INNOVATIONS 2004)*, W. Aung, R. Altenkirch, T. Cermak, R. W. King, and L. M. S. Ruiz, Eds. Arlington, VA: iNEER, 2004, pp. 93–106.

[9] A. Selmer, M. Goodson, M. Kraft, S. Sen, V. F. McNeill, B. S. Johnston, and C. K. Colton, "Performing process control experiments across the atlantic," *Chem. Eng. Educ.*, vol. 9, pp. 232–237, 2005.

[10] D. Talavera, "On-line laboratory for remote polymer crystallization experiments using optical microscopy," M.Eng. thesis, Massachusetts Inst. of Technology, Cambridge, MA, 2003.

[11] J. O. Campbell, R. J. Bourne, P. J. Mosterman, and J. A. Brodersen, "The effectiveness of learning simulators in electronic laboratories," *J. Eng. Educ.*, vol. 91, pp. 81–87.

[12] M. M. Albu, K. E. Holbert, G. T. Heydt, S. D. Grigorescu, and V. Trusca, "Embedding remote experimentation in power engineering education," *IEEE Trans. Power Syst.*, vol. 19, no. 1, pp. 139–143, 2004.

[13] S. Srinivasan, L. Pérez, R. D. Palmer, D. W. Brooks, K. Wilson, and D. Fowler, "Reality versus simulation," *J. Sci. Educ. Technol.*, vol. 15, no. 2, pp. 137–141, Apr. 2006.

[14] G. Klimeck, M. McClennan, M. Mannino, M. Korkusinski, C. Heitzinger, R. Kennell, and S. Clark, "NEMO 3D and nanoHUB: Bridging research and education," in *Proc. 6th IEEE Conf. Nanotechnol.*, Jun. 17–20, 2006, vol. 2, pp. 441–444.

[15] J. Travis and J. Kring, *LabVIEW for Everyone*^{3}. Upper Saddle River, NJ: Prentice-Hall, 2006.

[16] J. Travis, *Internet Applications in LabVIEW.* Upper Saddle River, NJ: Prentice-Hall, 2000.

[17] I. Foster and C. Kesselman, *The Grid: Blueprint for a New Computing Infrastructure.* San Francisco, CA: Elsevier, 2004.

[18] F. Berman, G. C. Fox, and A. J. G. Hey, *Grid Computing: Making the Global Infrastructure a Reality.* Chichester, U.K.: Wiley, 2004.

[19] F. Lelli and G. Maron, "The GRIDCC project," in *Distributed Cooperative Laboratories: Networking, Instrumentation, and Measurements*, F. Davoli, S. Palazzo, and S. Zappatore, Eds. New York: Springer, 2006, pp. 269–277.

[20] T. Devadithya, K. Chiu, K. Huffman, and D. F. McMullen, "The common instrument middleware architecture: Overview of goals and implementation," in *Proc. 1st IEEE Int. Conf. e-Science Grid Comput. (e-Science 2005)*, Melbourne, Australia, Dec. 5–8, 2005, pp. 96–106.

[21] R. Bramley, K. Chiu, T. Devadithya, N. Gupta, C. Hart, J. C. Huffman, K. L. Huffman, Y. Ma, and D. F. McMullen, "Instrument monitoring, data sharing and archiving using Common Instrument Middleware Architecture (CIMA)," *J. Chem. Inf. Model.*, vol. 46, no. 3, pp. 1017–1025, May–Jun. 2006.

[22] J. L. Hardison, D. Zych, J. A. del Alamo, V. J. Harward, S. R. Lerman, S. M. Wang, K. Yehia, and C. Varadharajan, "The microelectronics WebLab 6.0—An implementation using web services and the iLab shared architecture," presented at the Int. Conf. Eng. Educ. Res. 2005, Tainan, Taiwan, R.O.C., Mar. 1–5, 2005.

[23] K. Amaratunga and R. Sudarshan, "A virtual laboratory for real-time monitoring of civil engineering infrastructure," presented at the Int. Conf. Eng. Educ. 2002, Manchester, U.K., Aug. 18–22, 2002.

[24] J. Harward, J. A. del Alamo, V. S. Choudary, K. DeLong, J. L. Hardison, S. R. Lerman, J. Northridge, C. Varadharajan, S. Wang, K. Yehia, and D. Zych, "iLabs: A scalable architecture for sharing online laboratories," presented at the Int. Conf. Eng. Educ. 2004, Gainesville, FL, Oct. 16–21, 2004.

[25] J. Harward, J. Northridge, R. Zbib, L. Naamani, I. Jabbour, and T. T. Mao. (2006, May 11). *iLab Interactive Ticketing and Integrated Management—Overview.* Cambridge, MA: MIT Press. Available: http://icampus.mit.edu/iLabs/Architecture-/downloads/protectedfiles/Ticketing&Integrated-Management.doc

[26] R. A. Mitchell, J. Fischer, and J. A. del Alamo, "A survey study of the impact of a MIT microelectronics online laboratory on student learning," in *Proc. Int. Conf. Eng. Educ. (ICEE)*, San Juan, Puerto Rico, Jul. 23–28, 2006.

[27] J. A. del Alamo, R. Mitchell, and J. Fischer, "Working with WebLab: Assessment and the improvement in teaching and learning in an undergraduate course at MIT," in *Proc. NAE/CASEE Dane Mary Louise Miller Annu. Symp.*, San Diego, CA, Oct. 27, 2006.

[28] J. Fischer, R. Mitchell, and J. del Alamo, "Inquiry-learning with WebLab: Undergraduate attitudes and experiences," *J. Sci. Educ. Technol.*, vol. 16, no. 4, pp. 337–348, Aug. 2007.

[29] G. Hovland, "Evaluation of an inverted pendulum control experiment," *IEEE Trans. Educ.*, to be published.

[30] G. Viedma, I. J. Dancy, and K. H. Lundberg, "A Web-based linear-systems iLab," presented at the Amer. Contr. Conf. 2005, Portland, OR, Jun.8–10, 2005.

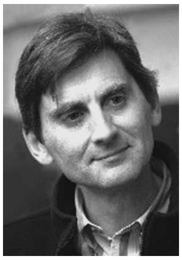
[31] S. C. Ehrmann, S. W. Gilbert, and F. McMartin, *Factors Affecting the Adoption of Faculty-Developed Academic Software: A Study of Five iCampus Projects*. Takoma Park, MD: TLT Group. [Online]. Available: http://www.tltgroup.org/icampus/

[32] J. A. del Alamo, V. Chang, J. Hardison, D. Zych, and L. Hui, "An online microelectronics device characterization laboratory with a circuit-like user interface," in *Proc. World Innov. Eng. Educ. Res. (INNOVATIONS 2004)*, W. Aung, R. Altenkirch, T. Cermak, R. W. King, and L. M. S. Ruiz, Eds. Arlington, VA: iNEER, 2004, pp. 153–162.

ABOUT THE AUTHORS

V. Judson Harward received the Ph.D. degree from Harvard University, Cambridge, MA.

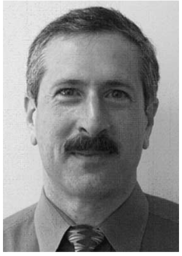
He is the Associate Director and Principal Research Scientist at the Center for Educational Computing Initiatives, Massachusetts Institute of Technology, Cambridge. He has been a Software Architect and Project Manager for the iLab Shared Architecture since its inception.



Jesus A. del Alamo (Fellow, IEEE) received the telecommunications engineer degree from the Polytechnic University of Madrid, Spain, in 1980 and the M.S. and Ph.D. degrees in electrical engineering from Stanford University, Palo Alto, CA, in 1983 and 1985, respectively.

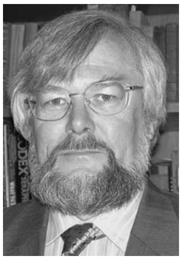
From 1985 to 1988, he was with NTT LSI Laboratories, Atsugi, Japan. He has been with the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology (MIT), Cambridge, since 1988, where he is currently Donner Professor and MacVicar Faculty Fellow. His current research interests are in microelectronics technologies for communications and logic processing. He is also active in online laboratories for science and engineering education.

Prof. del Alamo is a member of the Royal Spanish Academy of Engineering. He currently is Editor of IEEE ELECTRON DEVICE LETTERS. He has received several teaching awards at MIT: the Baker Award, the Edgerton Junior Faculty Achievement Award, the Smullin Award, the Bose Award, and the Donner Chair. He was a National Science Foundation Presidential Young Investigator.

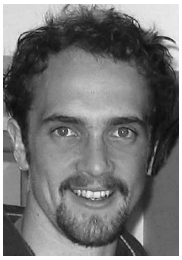


Steven R. Lerman received the B.S. and M.S. degrees in civil engineering and the Ph.D. degree in transportation systems analysis from the Massachusetts Institute of Technology (MIT), Cambridge, in 1972, 1973, and 1975, respectively.

He holds the Class of 1922 Distinguished Professorship at MIT. He was Chair of the MIT Faculty from 1999 to 2001 and interim Chair from 2006 to 2007. He is currently the Dean for Graduate Education, Acting Codirector of the Singapore-MIT Alliance, and Director of the Center for Educational Computing Initiatives, an MIT-wide research center devoted to studying the application of computational and communication technologies in education. Prior to this, he was Director of Project Athena, which brought desktop computing to all students at MIT. He is a member of both the Management Board of MIT Press and the Board of Directors of Cambridge Systematics, Inc. He joined the MIT Faculty in 1975 and is now a Professor of civil and environmental engineering. His current research interest is the development of the iLab software architecture to support remote operation of laboratory equipment.



Philip H. Bailey is a Senior Project Manager with the Center for Educational Computing Initiatives (CECI), Massachusetts Institute of Technology, Cambridge. He joined CECI in 1992 and has worked on numerous CECI projects. Previously, he designed user interfaces for Bell Atlantic Software Systems, multimedia authoring tools for Times-Mirror, and manufacturing software for Honeywell/Bull.



Joel Carpenter received the B.Eng. degree in computer systems and the B.Sc. degree in physics from the University of Queensland, Australia, in 2006, where he is currently pursuing the master's degree in electrical engineering (microwave telecommunications).

Kimberly DeLong received the B.Sc. degree in computer science from Smith College, Northampton, MA, and the master's degree in computer systems engineering from Northeastern University, Boston, MA.

She has been with the Center for Educational Computing Initiatives, Massachusetts Institute of Technology, Cambridge, since 1995. During this time, she has worked on many projects involving a broad range of technologies. She has been involved with the iLab project since 2002.

Chris Felknor is a Software Engineer who has specialized in large-scale distributed applications using Microsoft technology, most notably the Staples.com and Quill.com e-commerce Web sites. He has contributed Microsoft-specific experience and knowledge of corporate best practices to the iLabs development effort.

James Hardison (Member, IEEE) received the B.S. degree in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge.

He is a Research Engineer with the Center for Educational Computing Initiatives, MIT. He is involved with the management and development of online laboratories within the iLab shared architecture.

Bryant Harrison received the B.Sc. degree in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, in 2007, where he is currently pursuing the M.Eng. degree in electrical engineering and computer science.

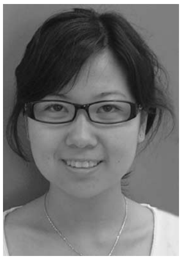
He led a team in Kenya teaching information technology and entrepreneurship in 2004 through the MIT Africa Internet Technology Initiative. He has coordinated the activities of the program in Africa since 2005. In 2006, he worked on the OpenAfrica project with Microsoft iCampus to assess the effectiveness of educational technologies in universities and high schools in East Africa. His current research interests include educational technologies and developmental entrepreneurship. His previous research and industry experiences include work in medical devices and bioelectronic instrumentation.

Imad Jabbour (Member, IEEE) received the B.Eng. degree in computer and communications engineering from the American University of Beirut, Beirut, Lebanon, in 2005 and the M.S. degree in information technology from the Massachusetts Institute of Technology, Cambridge, in 2007.

Since July 2007, he has been with Oracle USA, Inc., where he is currently a Product Manager with Oracle Financials.

Phillip D. Long received the Ph.D. degree from Pennsylvania State University, University Park.

He is an Associate Director in the Office of Educational Innovation and Technology. He led the iCampus Outreach effort, of which iLabs was a major educational technology initiative. He currently is Chair of the New Media Consortium Advisory Board.



Tingting Mao received the M.S. degree from Peking University, Peking, China in 2004 and the M.S. degree from the Massachusetts Institute of Technology (MIT), Cambridge, in 2007, where she is currently pursuing the Ph.D. degree in the Systems Program, Civil and Environmental Engineering Department.

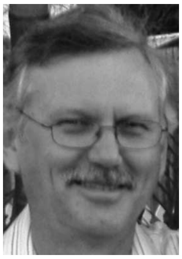
Her research interests include the interoperable Internet scale security framework for RFID Networks and privacy issues in pervasive computing. She was a Research Assistant with CECI/MIT.

Loai Naamani received the B.Eng. degree from the American University of Beirut, Beirut, Lebanon, and the M.Eng. degree from the Massachusetts Institute of Technology, Cambridge, where he is currently pursuing the Ph.D. degree in the Information Technology program.

He has helped integrate the Earthquake Shake Table WebLab with the iLab shared architecture.

Jedidiah Northridge received the undergraduate degree from Trinity College, Hartford, CT, and the master's degree from the Massachusetts Institute of Technology, Cambridge.

He is a Lead Software Engineer with ClearPoint Metrics, where he is working to build a software system that enables any company to practice quantitative information security risk management.



Mark Schulz received the Ph.D. degree from the University of Essex, U.K.

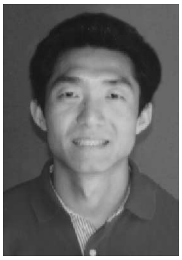
He is Head of the Higher Education, Research and Consultancy Unit, Tertiary and Educational Development Institute, University of Queensland, Brisbane, Australia. He leads the Australian Hub of the MIT iCampus Outreach project.

Daniel Talavera received the B.Sc. degree in biology and in electrical engineering and computer science and the M.Eng. degree in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, in 2002 and 2003, respectively. He received the M.Sc. degree in mechanical engineering from the Naval Postgraduate School, Monterey, CA.

His master's research involved mechanizing an interactive polymer crystallization experiment as part of the MIT iLab program. Following his graduate work at MIT, he joined the Naval Reactors Headquarters, Naval Nuclear Propulsion Program, Washington, DC. His professional contributions include work on radiological emergency response, prototype reactor plant fluid systems, and reactor safety analysis at Naval Reactors.

Charuleka Varadharajan received the B.Tech. degree from the Indian Institute of Technology, Chennai, and the M.Sc. degree from the Massachusetts Institute of Technology, Cambridge, where she is currently pursuing the Ph.D. degree in the Department of Civil and Environmental Engineering.

She helped develop parts of the iLab shared architecture, such as the database, APIs, and user interface.



Shaomin Wang received the Ph.D. degree in information technology from the Massachusetts Institute of Technology (MIT), Cambridge, in 2004.

He worked on the iLab Project while he was a Graduate Research Assistant at MIT. His research interests are information retrieval; semantic text analysis, and software engineering. He is now with Oracle Inc.

Karim Yehia was a Research Assistant on the iLab project from spring 2003 to spring 2004. His research focused on developing the iLab authorization model and extending the common services provided by iLab to interactive laboratories. He is now a Developer in financial technology with Deutsche Bank, New York.

Rabih Zbib received the B.E. degree (with distinction) from the American University of Beirut, Beirut, Lebanon, and the M.S. degree in information technologies from the Massachusetts Institute of Technology (MIT), Cambridge, where he is currently pursuing the Ph.D. degree in information technology.

His dissertation is in the area of automated machine translation. He has been a Research Assistant with MIT's Center for Educational Computing Initiatives, where he worked on the architecture, design, and implementation of the iLab architecture. Previously, he was a Research Scientist with the Applied Research Lab, Telcordia Technologies, for five years.

David Zych received the B.S. and M.E. degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge.

He was the primary Client Developer for the MIT Microelectronics Weblab from 2002 to 2007 and contributed significantly to the design of the iLabs architecture. He is now a Software Developer with the National Center for Supercomputing Applications, University of Illinois, Champaign.