MIT iLabs: Towards a Community of Internet Accessible Laboratories

Jesus. A. del Alamo, Philip Bailey, James Hardison, V. Judson Harward, Steven R. Lerman, Phillip D. Long, MIT

Sloan-C International Conference on Asynchronous Learning Networks

November 19, 2005

Sponsorship: Microsoft Corp., Carnegie Corp. of New York
Contents

- Motivation
- iLabs at MIT
- The iLab Shared Architecture
- Outreach - sharing technology, innovating education
Motivation to iLabs

- There is enormous educational value in hands-on laboratory experiences

- But, conventional labs…
  - … are expensive and have complex logistics
  - … can’t easily be shared

- iLabs (or “WebLabs”): real laboratories that are accessed through the Internet from anywhere at any time
iLabs at MIT

Dynamic signal analyzer
(EECS, deployed 2004)

Microelectronics device characterization
(EECS, deployed 1998)

Polymer crystallization
(Chem. E., deployed 2003)

Heat exchanger (Chem. E., deployed 2001)

Shake table (Civil Eng., deployed 2004)
MIT Microelectronics WebLab

Semiconductor Parameter Analyzer, Switching Matrix (donation of Agilent Technologies)

Device under test

Device test fixtures (donation of Agilent Technologies)

W2000 server
Typical Assignment

Transistor characterization project:

- Measure transistor characteristics
- Extract transistor parameters
- Compare measurements with class models

Also, do whatever else you want with the transistor.
Microelectronics device characterization:

- over 3000 student users (for credit) since 1998
“Formal” use of WebLab

Year

# students

- non-MIT (industrial)
- non-MIT (academic)
- MIT graduate
- MIT undergraduate

Year

2005 (projected)
- What is the lab system capacity?
- When do students carry out assignment?

Oct. 13-20, 2000

2PM: 6.012 exercise out (75 students)
4PM: 6.720J/3.43J exercise out (25 students)

2PM: 6.012 exercise due
4PM: 6.720J/3.43J exercise due
WebLab Capacity

System capacity: > 2,000 users/week, > 15,000 jobs/week

2PM: 6.012 exercise due
4PM: 6.720J/3.43J exercise due

[Oct. 13-20, 2000]
Shake Table

Goal:

Study behavior of building model structure to ground vibration

Relevance:

Earthquake building engineering
The iLab Vision:
the iLab Shared Architecture
The iLab Vision

- Many labs shared worldwide
- Some are unique (unreachable locations, rare materials)
- Many simple labs
The iLab Vision

- GUI to lab
- Integrates useful generic tools (graphing, numerical analysis, simulators)
- Allows for remote collaboration and tutoring
The iLab Vision

- Serves GUI to Client
- Mediates between Client and Lab Server
- Performs generic functions: user management, data storage
- Single account access to many labs
- Managed by end user University
• Service Broker acquires user data from University Databases
• User authentication through University IT infrastructure
The iLab Vision

- Order of magnitude more lab experiences
- More lab time to users
- More sophisticated labs available
- Communities of scholars created around iLabs sharing educational content
Conclusions

- iLabs will enhance science and engineering education
- iLabs and their educational content will be broadly shared around the world
- iLabs provide a path for the developed world to support education in the developing world
- iLabs Shared Architecture: scalable framework for iLabs, well suited to needs of developing world
The iLabs Architecture

A detailed look
iLab Design Goals

- Scaling usage of online labs to a large number of users
- Encouraging researchers and universities to share their labs online
- Single sign on to labs at multiple universities
- Freeing lab owner/operator from administration (i.e. authentication, authorization, storage of results, archiving of data, etc.) of users from other universities
- Allowing universities with diverse network infrastructures to interoperate and share resources
Project Boundaries

- Our architecture doesn’t deal with specific hardware and software interfaces to lab equipment.
- Our architecture is intended to be compatible and complementary with commercial software such as National Instruments LabView and analysis packages like Matlab.
iLab Generic Services

- User authentication (and registration)
- User authorization and credential (group) management
- Experiment specification and result storage
- Lab access scheduling
Lab Server

Internet

Client

- **Lab developer** responsible for 100% of development
  - Long time to deployment
- **Lab owner** responsible for 100% of management
  - The lab itself
  - User accounts, data storage, authentication, security
- **Students** need multiple accounts to access multiple labs
iLab Shared Architecture

Lab Servers → Internet → Generic iLab Service Broker → Clients

Internet

Generic iLab Service Broker

Campus network

Local databases

Campus network
The Case for Web Services

- Web services represent the latest version of an old concept -- they allow one computer to invoke a procedure (method) on another.
- They are platform and vendor independent (we have already successfully bridged a Java client ↔ a Windows XP/.NET Service Broker ↔ a Windows 2000 lab server (with NI GPIB)).
- Because they are usually based on http that we all use to access the web, they work well with campus networks.
- The iLab Shared Architecture builds on top of the current generation of web services.
iLab Experiment Typology, 1

3 Waves of Development

- **Batched Experiments (2003-2005):**
  - The entire specification of the experiment is determined before execution begins.
  - The user need not remain online while experiment executes.

- **Interactive Experiments (2004-2006):**
  - The student client portrays virtual lab equipment (GUI).
  - The student can interact with experiment throughout its course.
3 Waves of Development

- Sensor Experiments (2005-2007?):
  - Publish and subscribe based architecture
  - Triggers and event-driven data monitoring
  - Flexible data analysis
  - Data archive
iLabs Design Strategy

Separate responsibilities of the lab provider from those of the teaching faculty

- The lab provider designs and makes the laboratory experiment available online in as effective a presentation as possible.
- The teaching faculty register their own students, manage their accounts and result storage, and set course policy (e.g. can students collaborate).
Service Broker Responsibilities

The Service Broker is a domain independent server that

- stores and manages student experiment records;
- provides mechanism for but does not specify local campus course and privacy policy;
- authenticates users and transmits credentials *but not user IDs to Lab Server*;
- manages workflow between client and lab server
Lab Provider Responsibilities

The Lab Server team

- builds the lab server which must implement the web service methods that the Service Broker uses to forward experiments and retrieve results;
- usually supplies the student lab client software, which must implement the corresponding methods to allow the client to communicate with the Service Broker;
Student Web Session

1. User authenticates over SSL

2. SB lists user’s groups

3. User chooses effective group for session.

4. SB lists available Lab Clients

5. User chooses Lab Client for session.

6. SB launches client.
The student contacts the Service Broker (SB) via a standard web browser.

The student either

- registers for a new account, or
- authenticates himself to the Service Broker (current implementation offers ID/password over SSL)

The SB lists the student’s group (class) memberships, and asks the student to choose an effective group for this session.

The SB lists the lab servers/clients available to that effective group, and asks the student to choose a client

The SB launches the lab client (often an applet) for the student.
Service Broker: Launching the Client

My Clients

Messages for this Group:
The WebLab 6.0 Lab Server is available and operating normally.
Date Posted: 8/19/2004 11:02:45 AM

Lab Client: MIT Microelectronics Weblab

Version: 6.0 Graphical Applet
Description: The new Graphical client for Microelectronics
IMPORTANT: This client requires Java Plugin 1.4.2 in order, below) for details. Mozilla Firefox users must disable popup documentation.
Contact Email: use the "Report a Bug" page if you have problems.

Launch Client

Documentation View the Lab For Educators
Batched Experiment Submission

Web Service Calls

1. Lab Client submits an experiment

2. Lab Server receives submission

3. Service Broker processes submission
   - returns SubmissionReport containing experimentID

4. Lab Client receives submission report
Batched Experiment Result Retrieval
Web Service Calls

Lab Client
- RetrieveResult(experimentID)
- Notify(experimentID)
  - returns ResultReport

Lab Server
- RetrieveResult(experimentID)
  - returns ResultReport

Service Broker

1. Notify(experimentID)
2. RetrieveResult(experimentID)
3. returns ResultReport
4. RetrieveResult(experimentID)
5. returns ResultReport
Service Broker Administrative Services

- Adding, modifying, and removing lab servers and clients.
- Adding, removing, or confirming user access.
- User management including assigning users to groups and modifying access rights.
- Managing experiment records.
In the batched experiment architecture, the client and the lab server communicate only through the Service Broker:

Lab Client \(\rightarrow\) Service Broker \(\leftarrow\) Lab Server

Interactive Experiment

No Direct Communication
Shaketable Prototype

Major Milestone, 5/2005:

*The 1st Prototype iLab Interactive Lab*

- Uses the new iLab interactive authorization (ticket) architecture
- Does not disrupt the original implementation
iLabs Dissemination

iCampus Outreach
MIT iCampus Affiliates Program

- World-wide scalability
- Tiered model of engagement
- Leveraging community

“Our academic community is very excited about iCampus.”
Dr. Miguel A. Romero
Director, Environmental Quality, ITESM

“We strongly believe that the development of advanced pedagogies with information technology requires a considerable investment, which can only be sustainable if the investment is amortized over a number of institutions”
John Norman,
Director, CARET
University of Cambridge

http://icampus.mit.edu/ilabs
Supporting Adoption

- iCampus Outreach for iLabs
  - Hub and spoke model
  - Community support
  - Workshops
  - Online discussion forum
  - Seed grants to ease adoption barriers
iLab Partners Developer Support

- Developer visits
- Release of standard lab server and client modules
- VoIP conferencing
  - world-wide virtual development team
- Video conference virtual meetings
Adoption Case Study
University of Queensland

- iLabs Website
- Download iLab software components
- Install, study, & ask questions
University of Queensland
Beam Balancing Control Experiment
University of Queensland
Beam Balancing Control Experiment

1. Write MatLab Control Program
2. Upload to Server
3. Run Expt
4. Collect data & evaluate
5. Revise MatLab Control Program

11/19/2005
Beam Balancing - the Movie
Exploring iLabs

- Visit [http://icampus.mit.edu/ilabs](http://icampus.mit.edu/ilabs)
- Try out the public instance of WebLabs
  - [http://openilabs.mit.edu](http://openilabs.mit.edu)
- Download documentation and code
- Contact us at
  - icampus@mit.edu or longpd@mit.edu
- Talk to hub partners
  - Prof. Mark Schulz - ITEE, UQ, mschulz@uq.edu.au
  - Prof. Miguel Angel Romero Ogawa, mromero@itesm.mx
The Future of iLabs

The iLabs Foundation

Building a micro-economy of shared experiments
iLabs - Looking Ahead

iLabs Milestones & Future Directions

[Diagram showing milestones from 9/19/2005 to 12/29/2005, including events such as iLabs Hub Visit, UQ T&L Week, National Instruments LabVIEW Demo, and more.

11/19/2005
All MIT developed software has been and will continue to be made available for free under an open source license.

We encourage but do not require our academic partners to follow the same policy. The decision to share their code and under what terms is theirs to determine.

We allow industrial partners to develop commercial “shrink-wrapped” (supported) versions of the iLab components.
“If You Can’t Come to the Lab… the Lab Will Come to You!”

(Earth at 89 GHz; courtesy of J. Grahn, Chalmers U.)