

A Versatile Internet-Accessible Electronics Workbench with Troubleshooting Capabilities

[doi:10.3991/ijoe.v5s1.988](https://doi.org/10.3991/ijoe.v5s1.988)

H. Soumare, R. Shroff, J. L. Hardison, J. A. del Alamo, V. J. Harward, P. H. Bailey and K. K. DeLong
Massachusetts Institute of Technology, Cambridge, U.S.A.

Abstract—The MIT iLab Project was established to expand the range of laboratory experiences available to students in science and engineering education. iLabs are online laboratories that enable students to conduct real experiments remotely. Recently, the iLab Project has focused on building remote laboratories around the NI-ELVIS platform, an all-in-one electronics workbench. This paper will detail our recent efforts in expanding the capabilities of ELVIS-based iLabs by enabling students to test and debug digital and analog circuits. This work will enable students to perform remote experiments characterizing digital logic elements. By merging switching capabilities with the Digital Multimeter available on the ELVIS, students will have the ability to examine and troubleshoot circuits. These added capabilities will provide educators and students with unparalleled flexibility and significantly enrich the remote laboratory experience.

Index Terms—e-learning, electrical engineering, ELVIS, iLab, remote laboratories

INTRODUCTION

A student's curiosity is often left unsatisfied with a classroom education. In many cases, science and engineering courses are taught with an accompanying laboratory component. These laboratory experiments provide students with important hands-on, practical experience to give more credibility to the theories and methods taught in class. Further, they equip students with a variety of skills necessary for them to be successful in the real world. A purely theoretical education would shield students from the inevitable discrepancies they would face when applying their education as professionals. These inconsistencies are often a result of noisy data, inefficiencies of components and sometimes even a failure to setup the problem correctly. A practical laboratory component provides students with experience handling real data and enables them to deal with these issues in an educational setting so that they can overcome these problems more competently as professionals.

However, students often don't have sufficient access to traditional, hands-on laboratories. This is because traditional lab facilities are expensive to set up and maintain. Further, large class sizes and limitations in class

time and equipment availability have led to many institutions around the world neglecting this necessary component of their students' education.

Moreover, traditional laboratories, where available, rarely reach their full potential. This is because, for the most part, their access is limited. Traditional laboratories are generally open only when staffing is available and there are necessarily a limited number of experimental setups for students to use. This presents a considerable inconvenience for students, who are often so engaged in assembling and troubleshooting an experiment that they are not left with enough time to absorb the crucial concepts the assignment was designed to demonstrate.

In 1998, MIT created the concept of iLabs - an attempt to bridge the challenges and inefficiencies of using traditional laboratories [1]. iLabs are online laboratories that give students access to various experimental setups thus enabling them to conduct experiments on real equipment remotely via the Internet. Such a setup bypasses a large number of the typical problems of conventional laboratories. Most importantly, iLabs greatly reduce the cost of setting up a laboratory for each user because only one piece of equipment, which can be shared by multiple users, is required. Further, iLabs do not require direct staffing while they are being used. As a result, an iLab can be setup with an experiment, placed in any location where there is Internet access and made available without time restrictions. Thus, students do not have to wait in long queues to use equipment because they can access the setups at their convenience from their own homes. Further, since iLabs can provide access to laboratory instrumentation to a broad range of users over the Internet, only a few institutions need to invest in costly equipment. This scalability of the iLabs framework lends itself to the project's vision: A global network of institutions creating and sharing cross-disciplinary iLabs for use by each others' students and faculty.

Since then, the MIT iLab Project has focused on the design of a common architecture for the development and deployment of online laboratories called the iLab Shared Architecture [2]. The iLab Shared Architecture, or ISA, provides a set of generic lab services, such as user account management and data storage, in a middleware system that can be accessed using Web Services. In this architecture, iLabs can be designed as distributed web applications, rather than stand-alone systems, with the ISA middleware acting as a distributor of remote laboratories to users. In this way, the ISA not only eases the deployment of remote laboratories, but also makes them easier to share across institutions. Adoption of the iLab Shared Architecture has led to faculty from a variety of engineering fields implementing and deploying a number

This work has been supported in part by the Carnegie Corporation of New York, the Maricopa Advanced Technology Education Center, the Microsoft Corporation through iCampus (the MIT-Microsoft Alliance), the National Science Foundation under award #0702735, by the Singapore-MIT Alliance, the Singapore-MIT Alliance for Research and Technology and by MIT Alumni Funds (Classes of '51, '55, '60, and '72) as well as through equipment donations from National Instruments, Agilent Technologies, AMD, Hewlett-Packard and Intel.

of iLabs which can be accessed by students across the globe [3-5]. During this time, the number and variety of iLabs available to a given user has expanded significantly [6].

Within the MIT iLab Project a significant amount of effort has been applied to developing iLabs based on the National Instruments Educational Laboratory Virtual Instrumentation Suite (ELVIS) platform [7]. The ELVIS is a relatively inexpensive, small-footprint instrument containing a number of common electronics test and measurement tools [8]. Leveraging iLabs together with the ELVIS hardware platform enables a wide variety of electronics measurements to be made remotely without the burden of interfacing with many different pieces of individual lab instruments. Initially, ELVIS-based iLabs were focused on a small subset of the platform's functionality. More recently, the iLab-available feature set has grown, enabling a truly versatile remote laboratory experience.

This paper will detail the latest additions to the ELVIS-based iLab developed by the MIT iLab Project. It will provide an overview of previous work with the platform. With the development trajectory established, this paper will describe the project's latest efforts to enable the entire set of ELVIS functionality available within a single iLab.

I. THE ILAB SHARED ARCHITECTURE

Early remote laboratories at MIT were built as individual, stand-alone systems [9-10]. A lab client, such as a Java applet, communicated directly with a server connected to the hardware being investigated. This server also provided an administrative interface, managing student accounts and logins. However, as the iLab project grew and more labs were created, this approach led to a divergence in development efforts with each lab having its own unique code structure. The iLabs shared architecture was introduced in 2002 in an effort to create a

standardized infrastructure which could be used by developers around the world [2].

As detailed in Fig. 1, the iLab architecture is split into three major components: a lab client, a Service Broker and a lab server which interact with each other using web-based service calls. Each of these components is described in detail in this section.

A. Lab Client

The lab client is the user interface through which students access an iLab. It provides an intuitive representation for the lab being run, allowing users to specify parameter values and to graph the results returned from a given experiment.

The current family of lab clients was adapted from the versatile, open source client used in the original Microelectronics Device Characterization iLab [3]. Thus, users of previous generations of MIT iLab clients will be familiar with the more recent lab clients.

B. Service Broker

The Service Broker serves as the heart of the ISA. A Service Broker provides generic administrative services that can easily be leveraged by lab developers while also managing communication between lab servers and clients. Given its generic design, a Service Broker can be used by multiple lab servers and clients simultaneously. In terms of administrative functionality, the Service Broker deals with user accounts and enforcing permissions. This allows administrators to group students by class, year, or even educational institution, for example, and to specify which labs are available to each group of students. The Service Broker also stores usage data that completely describes a lab session whenever a student runs an experiment.

There are two distinct types of Service Brokers, those for batched experiments and those for interactive ones. The Batched Service Broker supports the running of

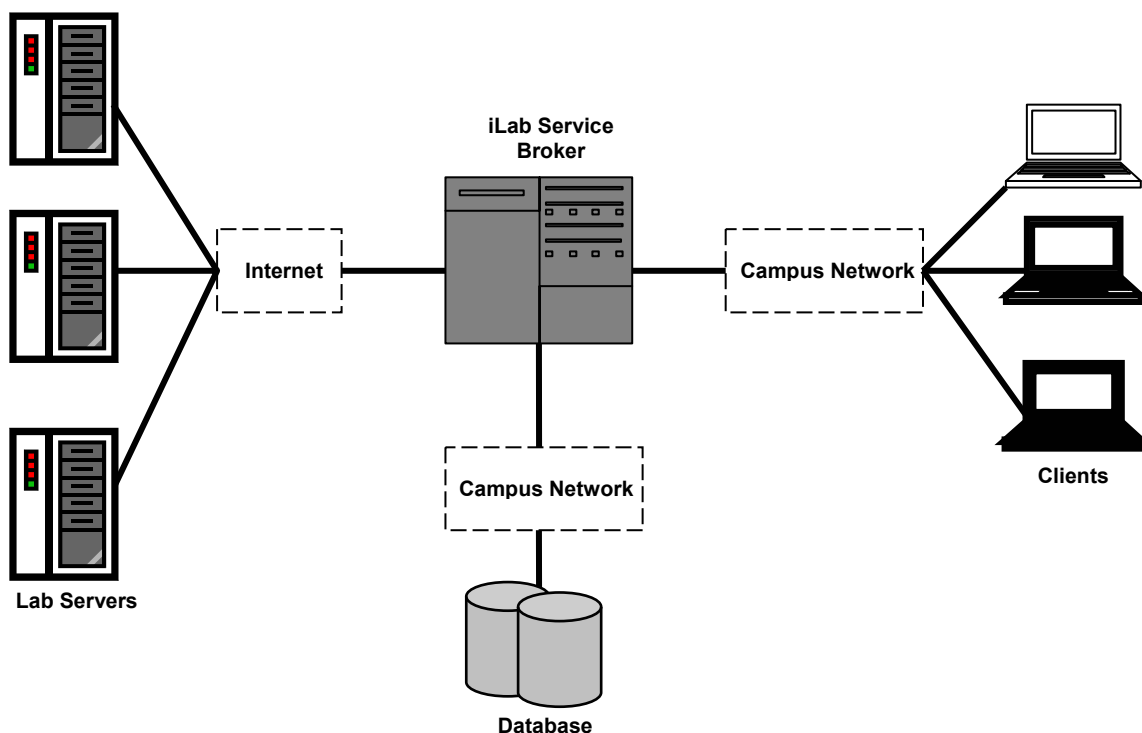


Figure 1. Topology of the iLab Shared Architecture

batched labs. A batched lab experiment is completely defined by specifying parameter values for the setup. This information is passed through the Batched Service Broker and stored in a first-in first out (FIFO) queue on the lab server. Once the experiment reaches the front of the queue, it is run and the results are passed back to the client. When running a batched experiment, the student does not have real time control of the lab equipment and, as a result, bandwidth requirements are quite low. This type of experiment is suitable in variety of cases [3,4,7,11].

On the other end of the spectrum is the Interactive Service Broker, which facilitates the running of interactive labs [5]. In an interactive lab, the student has complete and exclusive control of the experimental setup for a given period of time. This is a much more realistic simulation of the laboratory experience but presents several challenges such as the need for scheduling when dealing with large class sizes. In order to facilitate real-time control, Interactive iLabs are built in such a way that the lab client connects directly to the lab server. The Interactive Service Broker still handles user accounts and lab access, but once access is established, the Service Broker allows the lab client and lab server to communicate directly. This can enable a much more responsive experience, but at the cost of much higher bandwidth requirements.

The remainder of this paper focuses on iLab development within the iLab Shared Architecture for batched experiments.

C. Lab Server

The Lab Server is the part of the ISA which interacts directly with laboratory equipment. Specifications for each experiment, entered in the client by students, are sent to the lab server via the Service Broker. Analogously, once the experiment is performed the lab server sends the results back to the client through the Service Broker.

Another important function of the lab server is to serve as an administrative interface to the lab equipment. Lab administrators enter the desired configuration for each experiment by choosing which instruments they wish to include in a particular setup. Administrators are also given the opportunity to specify the allowable range of input values for each required parameter. Any values entered by the user outside this range are rejected. This ensures the safety of laboratory equipment and helps prevent potentially hazardous situations.

To facilitate the sharing of labs between institutions each lab server can interact with several Service Brokers.

It must, however, be noted that each lab server is equipment specific, and as such, a different lab server must be used for each distinct set of lab instrumentation.

D. Dataflow within the iLabs Shared Architecture

Lab-specific information is exchanged between the various components of the ISA using a series of XML encoded messages. These messages capture the lab configuration, the specification of a given experiment and the results of an experiment. One complete run of an iLab experiment involves the creation, transfer and deciphering of each of these documents. This exchange is depicted in Fig. 2.

Initially, when a user selects the lab he wishes to run in the Service Broker, a lab configuration message, or document, is created by the lab server and sent via the Service Broker to the client. This document encapsulates information about which lab resources are available in the present setup. On parsing the lab configuration document, the client retrieves the data it carries and displays the appropriate system-under-test and available instruments.

Once the assignment is displayed to the student, he configures the various instruments visible in the client by inputting appropriate parameter values. These values are compiled into an experiment specification document, which is sent to the lab server where it is parsed and translated into instrument specific commands. Once the experiment is conducted as per the user's specifications, the results are written into an experiment result document. This document is then sent to the client where the relevant information is extracted and graphed.

II. PREVIOUS WORK AND MOTIVATION

The National Instruments Educational Laboratory Virtual Instrument Suite (ELVIS) is an all-in-one device containing a suite of 12 instruments allowing students to perform hands-on experiments in electronics [8]. The ELVIS contains many of the essential instruments found in traditional electrical engineering laboratories, such as function generators, an oscilloscope, power supplies, a multimeter and digital I/O. These instruments are contained in a single unit with a circuit prototyping board connected to it. This combination of instruments allows for a wide variety of experiments to be performed using a single hardware platform.

Instruments on the ELVIS can be accessed using the knobs and switches on its front panel. The ELVIS is also highly customizable using the LabVIEW programming environment, allowing the instrument's features to be controlled easily with a computer. This enables the ELVIS to fit nicely into the iLab context.

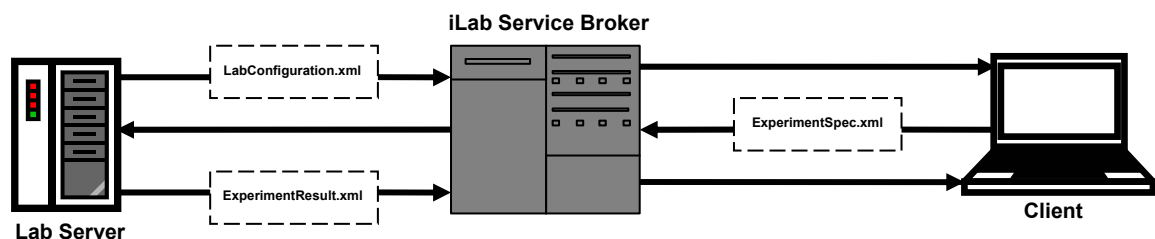


Figure 2. Dataflow within the iLab Shared Architecture

Since the adoption of the ELVIS as a cost-effective platform for the development of iLabs in 2006, there have been several, distinct releases of lab server and client software used to leverage the ELVIS platform for use as an iLab. Each of these versions has provided faculty with greater flexibility in designing meaningful assignments for use in engineering curricula. With every revision, particular attention was paid towards overcoming limitations of previous versions of the ELVIS-based iLabs, as exposed by students and faculty alike. The following section details the ELVIS iLab development trajectory (Fig. 3).

A. ELVIS iLab v1.0

The first version of the ELVIS iLab was completed in 2006 by Samuel Gikandi [12]. Gikandi used the Microelectronics iLab's general framework as a basis for its development. However, to make ELVIS-based iLabs more natural to use, a specifically tailored client was developed. The lab server was also appropriately modified to effectively communicate with the new hardware while the original Service Broker was used. ELVIS v1.0 exposed the Oscilloscope and Function Generator functions of the ELVIS, enabling users to study single input circuits.

Owing to the fact that only two of the ELVIS's instruments were exposed in v1.0, it did not support a very wide range of experiments. More specifically, the ELVIS v1.0 iLab could only handle time-domain experimentation. No frequency- or digital-domain analysis could be performed. This was a considerable limitation of this version, given the fact that these concepts form an integral part of most undergraduate level electrical engineering courses. Almost all iLabs using ELVIS v1.0 involved the characterization of individual electronic devices or small analog circuits with a single input/output configuration. Further, due to its inability to handle more than one input and output, or switch between existing setups, v1.0 only supported a single device/circuit configuration per board. This meant that if several instructors wished to use the ELVIS iLab as part of their

courses over a particular period of time, separate sets of hardware would be needed for each course. This, in a way, belied the ELVIS's reputation as a low cost replacement for higher end equipment. Another implication of the inability to switch between points on the ELVIS prototyping board was that every setup was fixed, i.e., a user could only analyze a previously wired circuit. For example, if he wished to measure input voltage instead of output voltage, it would not be possible. This would require the physical circuit to be rewired by an administrator. In other words, a student had absolutely no control over the apparatus.

Despite all these limitations, the ELVIS iLab v1.0 was a great success. iLabs based on this version were extensively used at MIT and other universities. This version also provided a robust platform conducive to subsequent development.

B. ELVIS iLab v2.0

Version 2.0 of the ELVIS iLab addressed some of the limitations present in ELVIS 1.0 by exposing more of the ELVIS functionality, namely:

Variable Power Supplies that output a DC voltage between -12 V and +12 V enabling some ability to control circuit operation;

Arbitrary Waveform Generator that allowed students to explore the response of circuits to user-defined waveforms;

Bode Analyzer that performs frequency-domain, swept sine measurements of analog circuits.

The inclusion of these instruments in an ELVIS-based iLab enabled a much broader set of measurements to be performed remotely [13,14]. On top of the ability to perform time-domain measurements with a basic function generator, students could provide a wider variety of input signals to a circuit-under-test or observe the frequency-domain behavior of that circuit. This provided a more complete laboratory experience for students and instructors.

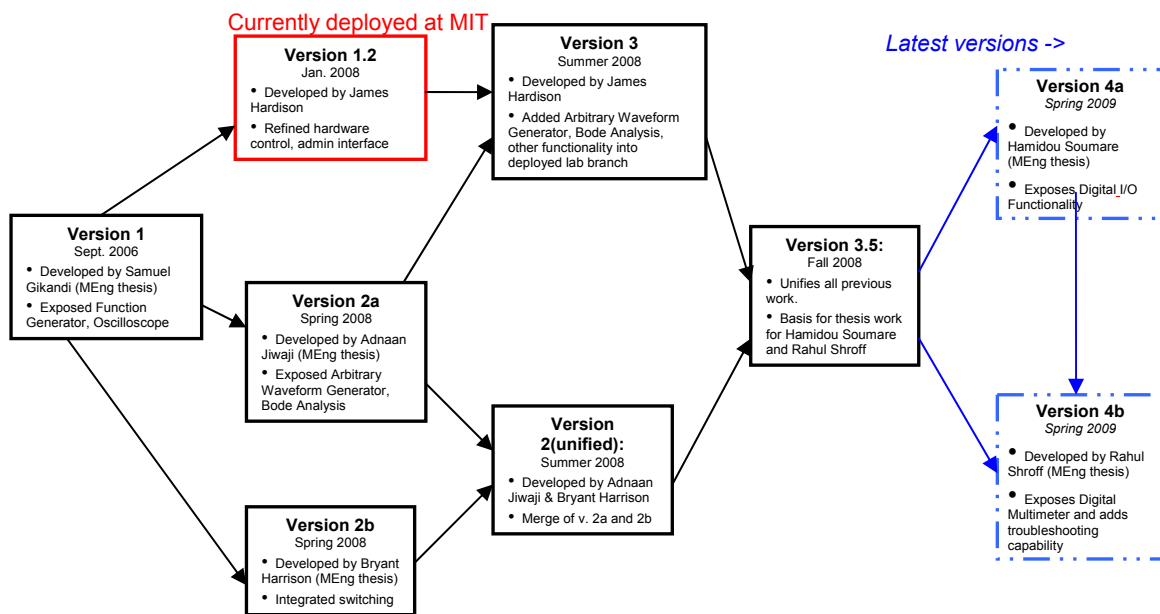


Figure 3. Development Trajectory for MIT ELVIS based iLabs

In addition to the functionality described above, v.2.0 of the ELVIS iLab provided the ability to switch components in or out of a circuit-under-test [13]. This work was inspired by a similar effort to leverage switching in electronics experiments by a partner iLab development group at Obafemi Awolowo University, Nigeria [7,11]. The MIT implementation used a LabVIEW controllable switch module in tandem with the ELVIS. The ability to switch components in a circuit provides a richer pedagogical experience since it allows students to experiment with elements of circuit design. In addition, it permits lab administrators to put numerous setups on a single ELVIS station allowing different courses to use the ELVIS equipment simultaneously.

C. Unification of Parallel Development Efforts in the iLabs Project (ELVIS iLab v3.5)

Branches in the development effort led to the emergence of two separate versions of the ELVIS iLab, each with its own functionality and distinct approach to the underlying LabVIEW Virtual Instrument (VI) structure. For example, James Hardison's v3.0 revision, intended to add some v2.0 functionality to the deployed ELVIS iLab branch (v1.2), did not incorporate any switching capabilities. However, this version used higher level ExpressVIs to drive the hardware, as opposed to the other versions which used lower level virtual instruments. Thus, to prevent further divergence and inconsistencies, the creation of a consolidated codebase was imperative. This new version would enable instructors and developers to use the entire breadth of exposed ELVIS functionality to create new and more meaningful labs.

To enable a greater variety of experiments to be run on the ELVIS iLab, it is crucial to increase the number of ELVIS instruments available through the iLab framework. The major contributions in the past involved the addition of a new "Domain" to the iLabs platform. The v1.0 ELVIS iLab set the foundation for the Time Domain. ELVIS iLab v2.0 introduced the Frequency Domain, our work on the v4.0 ELVIS iLab was designed to contribute the Digital-Domain and add troubleshooting capabilities (via the addition of the Digital Multimeter instrument) to EVLIS-based iLabs.

III. ELVIS v4.0 CONTRIBUTIONS TO THE ILABS PLATFORM

The new functionality available in ELVIS v4.0 provides several major contributions to the iLabs platform. First, the utilization of the ELVIS's digital I/O capabilities will lay the foundation for a whole host of remote experiments involving logic gates, programmable arrays, ROMs, FPGAs, and various other microcontrollers. Good understanding of digital electronics is essential for engineers in this increasingly digital world.

Second, v4.0 of the ELVIS iLab features the ELVIS platform's digital multimeter. Similar to a traditional handheld multimeter, this instrument enables users to easily measure current or voltage across a branch of a circuit as well as the values of standard electronic components (resistors, capacitors and inductors). In the ELVIS iLab, this instrument leverages the switching capability introduced in v2.0 in such a way that users can remotely specify what circuit branch or component to measure. This enables a remote DC measurement

experience approaching the flexibility of a similar traditional lab. Further, this functionality will enable instructors to use the v4.0 ELVIS iLab as a tool for teaching electronics troubleshooting skills.

Owing to the deliberate compartmentalization of components in the ISA, the process of adding digital I/O capabilities and the Digital Multimeter instrument into the already existing framework was very modular. The entire operation was extremely iterative, with several preliminary versions being tested and subsequently fine-tuned to provide the optimum user experience.

A. Integrating the Digital I/O Capabilities of the ELVIS

As discussed earlier, the lab server communicates directly with the lab instrumentation. At the heart of the effort to integrate digital logic analysis into the ELVIS iLab is the LabVIEW control software, which was designed with flexibility in mind.

Proper understanding of any digital circuit requires the creation of a truth table. A truth table describes the behavior of a digital logic element by specifying the appropriate output values for all possible input value combinations. If one were to fill out a truth table by experimentally observing a logic element, each permutation of possible input combinations would have to be tested. For example, an experiment involving three inputs has eight distinct combinations of inputs and would therefore require eight experiment runs to completely specify the corresponding truth table. The number of experiments required is exponentially related to the number of inputs. Given that we are dealing with a batched iLab, performing a large number of experiments is inconvenient and very time consuming. In response to this limitation, the LabVIEW digital I/O control software was designed to support input values that can change as a function of time. This allows the user to specify the digital input for up to eight clock steps, enabling the creation of a three input truth table in one experiment run.

With respect to the lab client, there were a number of changes required in order to support digital logic experiments. Of particular importance was upgrading the ELVIS iLab client applet's existing data graphing mechanisms. The existing client's graphing engine was designed to support plotting continuous data vectors on a Cartesian plot. In order to display digital I/O signals, the existing graphing engine had to be extended.

In order to graph data a user must specify data vectors for the x-axis and a y-axis, resulting in pointers to the specified data vectors within the graphing engine. Together, these data vectors specify a set of coordinate pairs representing the results. The client then draws the graph by looping through the set of coordinate pairs and connecting adjacent points by a line. In order to understand the limitations of the previous graphing engine, it is important to look at an example where it produces an undesired output.

Fig. 4 details the result of plotting a non-continuous data vector representing two digital lines. The result is almost as desired except for the diagonal line that connects the waveforms for two separate digital lines. The client uses a data type called a *connectpattern* that specifies a function for determining which points to connect in the graph. This data type returns true for the index of every point that should be connected to the

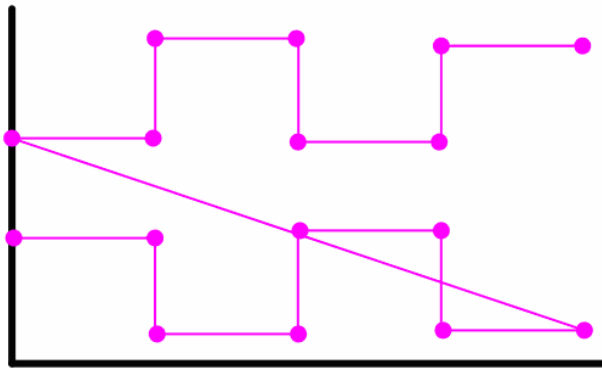


Figure 4. Sample stacked strip chart displaying the limitations of the previous graphing engine when plotting non-continuous data.

preceding point. The default *connectpattern* behavior is to “always connect”. This instructs the graphing engine to connect every adjacent point in the data plot, maintaining proper behavior for typical continuous data vectors. In order to properly display digital data, all adjacent points must be connected except for those belonging to different digital lines.

The ability to dynamically create and activate *connectpatterns* represents a significant addition to the iLabs platform. With this new behavior, developers are free to implement unique interpretations of the graph concept by creating functions that specify how to connect specific data points.

B. Enabling Troubleshooting Experiments Using the ELVIS's Digital Multimeter

The development process incorporating the Digital Multimeter into the ELVIS iLab framework involved considerable changes to both the lab server and the client. These were interfaced to a standard Batched Service Broker.

1) Lab Server Instrument Control

As an initial step towards integrating the Digital Multimeter the ELVIS-based iLab, LabVIEW control software was developed to facilitate interaction with the ELVIS hardware. The control software for the Digital Multimeter, or DMM, was designed to support the full set of DMM functions in the most modular way possible. Essentially, the DMM control module allows a calling method to input a measurement type in order to perform the desired measurement on the ELVIS. For each measurement type, the DMM control module selects the appropriate lower level parameters, enabling current, voltage or component value measurements (resistance, capacitance or inductance) using a single, simple program interface.

Once the DMM module had been created and tested thoroughly, it had to be integrated into the larger framework of the iLab ELVIS LabVIEW control software. The exact nature of this integration was influenced by both technical constraints and pedagogical decisions. From the technical perspective, certain ELVIS instruments make use of common hardware resources on the platform. For instance, the Digital Multimeter on the ELVIS could not be used simultaneously with the Function Generator as they use common lower level hardware resources [15]. This required that certain ELVIS

instruments be compartmentalized into experiment modes, thus preventing resource conflicts.

Fortunately, this decision was further justified by experiment pedagogy. Particularly, a broad set of experiments are possible when using the instruments built into the ELVIS. One can perform time-domain measurements using the Function Generator and Oscilloscope. Alternatively, one can also use the Bode Analyzer to perform frequency-domain measurements or the Digital Multimeter to observe DC measurements. While these may all be valuable, they are distinct types of measurements. Thus, compartmentalizing the individual ELVIS instruments into non-conflicting sets gave us the ability to organize those instruments into their appropriate measurement type.

2) Integration with Other Lab Server Modules

At a higher level in the lab server, several changes were made to supporting methods/routines in the lab server to recognize and take advantage of the Digital Multimeter. Perhaps the most interesting and useful change was the incorporation of an augmented version of switching into the system.

The DMM control module allows users of ELVIS iLab v4.0 to take a wide array of elementary, point-to-point measurements. However, due to the nature of iLabs, students are only exposed to a diagrammatic representation of the circuit schematic they are examining. They do not have the opportunity to physically change the actual configuration of a circuit. Given this limitation, the probes of the DMM had to be connected to two fixed points in a circuit. Using this model, students would only be able to take measurements across two previously selected circuit points. In order to understand this, consider the simple RLC circuit seen in Fig. 5. In the existing model, an administrator would connect the DMM across, for instance, the capacitor and students would only be able to make voltage, current or component measurements across that portion of the circuit. If a student wanted to find the value of R as labeled in the figure, it would be impossible without physically reconfiguring the circuit.

We chose to expose the ELVIS's Digital Multimeter functionality with the aim of providing students with remote testing and troubleshooting capabilities previously

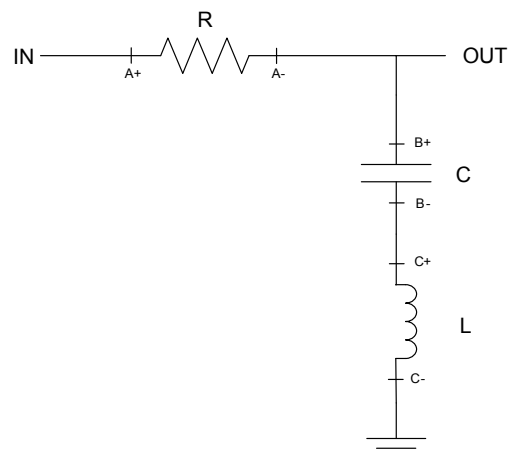


Figure 5. Simple RLC Circuit

unavailable on ELVIS-based iLabs. This implies that students should have complete flexibility in dynamically changing the location of the multimeter probes in a circuit. This would, to a certain degree, replicate what he or she would do while examining a circuit in a traditional laboratory setting.

To overcome this limitation of single point investigation, we combined the switching introduced by Harrison in the ELVIS iLab v2.0 with the DMM functionality. Due to the fact that v2.0's switching focused primarily on switching components into a circuit rather than instrument probes, it needed to be modified for use with the Digital Multimeter.

To facilitate switching within the DMM framework we introduced the concept of *switching layers*. There are four switching layers that assist with DMM measurements:

1. The first layer consists of switches defining measurement category. This level is necessary because the ELVIS uses different physical terminals to conduct what it calls 'Current Type' and 'Voltage Type' measurements.
2. The second layer of switching is related to measurement position. This is the layer of switching which is used to specify where exactly the user wishes to place the multimeter probes in a circuit. These correlate with points of interest which are marked on the circuit schematic displayed in the client to the user. Points labeled with a "+" or "-" indicate connections for the DMM's positive or negative terminals respectively
3. A third layer of switching is used to facilitate current measurements. Typically, while taking current measurements, the branch being investigated is temporarily disconnected from the circuit. The terminals of the DMM are then used to bridge the resulting open circuit so that the current can flow through the instrument.
4. The fourth layer of switching deals with peripheral instruments. These are switches connected to various sources and grounds which are part of the circuit. Correctly switching these components is required in order to perform accurate measurements.

To conduct a measurement, a user will specify only the points in the circuit where the multimeter probes should be placed along with the type of measurement to be made. Based on these parameters and the requirements of the *switching layers*, the lab server toggles the necessary switches in order to produce the desired connection between the Digital Multimeter and the circuit-under-test. In this way, the student can remain focused on the operation of the multimeter while the lab server manages the state of each individual switch

3) Lab Client Development

A considerable amount of time was spent making the Digital Multimeter-enabled iLab more user friendly. The DMM provides students with a slightly different user experience as compared to other instruments on the ELVIS. For the first time, users of the ELVIS-based iLab are given flexibility regarding where in the circuit-under-test they wish to connect an instrument.

In addition to making iLabs involving the Digital Multimeter natural to use, the client's design was greatly influenced by the need to always present an electrically

accurate circuit to the user. Consider the circuit schematic seen in Fig. 5. The added switching functionality enables users to choose which points ($A+$, $A-$, $B+$ etc) to investigate on this circuit. However, a student is completely oblivious to the machinery used to achieve this. Due to this, the schematic seen in Fig. 5 might be potentially confusing for students. For example, the wire connecting one end of the resistor to the capacitor in the circuit represents *one* electrical node. However, to facilitate flexibility in measurement there exist *two* distinct points ($A-$ and $B+$) on this wire to which the DMM terminals can connect. From the DMM point of view, having these two points is absolutely necessary since we must be able to physically connect both its positive and negative terminals to an electrical node for analysis. However, since the user only sees a virtual representation of the circuit and is unaware of the switching mechanism used to take multipoint measurements, seeing nodes labeled in this way might be disconcerting.

Doing away with these point labels would greatly reduce the flexibility of taking measurements with the DMM. Users would now only be able to take measurements across points previously selected by an administrator. To prevent this, we chose to represent these electrical nodes as distinctly colored clusters. As seen in Fig. 6 each cluster consists of three elements (a '+' and '-' representing connection points for the DMM's virtual probes, and a colored circle representing the actual node) which represent an electrical node. This approach preserves the inherent usability of the system while making node locations clear. To make it even more clear to students, once the DMM has been configured in the client, its icon morphs to reflect which points on the circuit it is connected to and also what type of measurement it is taking (Fig. 6).

Thus, using the machinery described above, a user can not only choose which type of measurement they want to make, but can also select where in the circuit-under-test to take that measurement. Thus, the Digital Multimeter

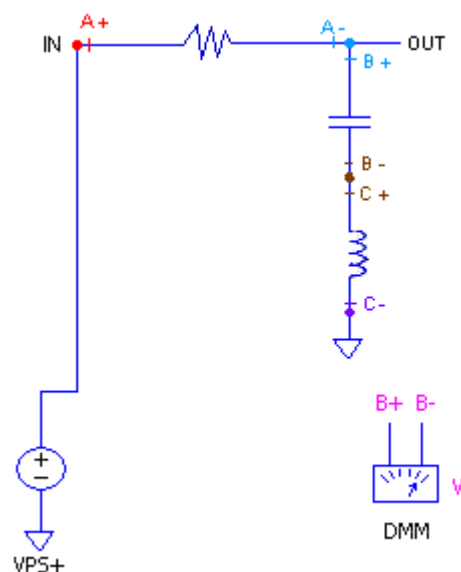


Figure 6. RLC Circuit with lab client rendered instrument terminals. Electrical nodes are represented by colored clusters.

provides students with real-time, dynamic circuit testing and debugging capabilities, unprecedented in an iLab. This significantly enhances the ELVIS iLab's value as a versatile educational tool and represents a considerable step forward in bridging the gap between conventional and remote laboratories.

IV. CONCLUSIONS

The ELVIS v4.0 iLab, by supporting digital logic as well as DC and component measurements through the Digital Multimeter, has greatly improved the value of the ELVIS-based iLab as a tool for electrical engineering instructors. Since integrated circuits are the backbone of modern day computing, the incorporation of digital capabilities will create a new class of user of the ELVIS v4.0. Diversifying the target audience for iLabs is a very important step towards realizing the project's overall goal of increasing the number and variety practical experiences available to students through remote online laboratories.

Detractors of the iLab concept have always been quick to point out that, using iLabs, students are unable to test or troubleshoot a system-under-test. Although conceived as a complement to hands-on laboratory exercises, given their cost-effective nature it is inevitable that in certain situations iLabs will be used to completely replace their traditional counterparts. In these situations students using earlier versions of ELVIS-based iLabs would never be exposed to these invaluable skills. The ELVIS iLab v4.0 takes a step towards overcoming this disability by introducing the Digital Multimeter and coupling its use with an augmented version of the already existing switching capabilities. However, the multimeter's use is not limited to debugging. A wide range of laboratory assignments, ranging from simple experiments involving series resistance to much more complex circuits can be designed around this instrument.

These developments significantly enhance an iLab's value as a versatile educational tool and represent a considerable step forward in bridging the gap between conventional and remote laboratories.

V. FUTURE WORK

ELVIS-based iLabs have greatly matured since their introduction in 2006. Constant development has seen a two instrument setup trying to mimic the Microelectronics Device Characterization iLab transform into a compelling platform supporting measurements of both digital logic systems and analog circuits in the DC-, time- and frequency-domains along with troubleshooting mechanisms.

However, in a bid to add new functionality some aesthetic details have been overlooked by developers. Although we have done some work in trying to improve circuit representation, what the iLab client lacks is the ability to change the way a circuit looks in real-time. For example consider the physical changes in a circuit that occur when taking measurements with the Digital Multimeter. Ideally, these changes, such as sources being shorted when taking component measurements or branches being opened when taking current measurements, should be made visible to users on the client's schematic panel. A further improvement would be to allow the position of the DMM icon to dynamically change based on a user's input. Changes of this nature will

require a significant redevelopment of the ELVIS lab client.

Additionally, National Instruments has released a much enhanced version of the ELVIS platform, called the ELVIS II. This new workbench provides more accurate measurements at low voltages - a major limitation in the original ELVIS - and has a frequency range much greater than its predecessor. Most importantly, it uses an entirely different underlying circuitry that prevents many of the resource conflicts inherent in the original ELVIS. An improvement to the existing ELVIS iLab implementation would be to make it compatible with this new hardware. This would enable remote circuit analysis with greater accuracy over a greater measurement range on a still-reasonably inexpensive hardware platform.

REFERENCES

- [1] J. A. del Alamo, L. Brooks, C. McLean, J. Hardison, G. Mishuris, V. Chang, *et al.*, "The MIT Microelectronics WebLab: a Web-Enabled Remote Laboratory for Microelectronics Device Characterization", 2002 World Congress on Networked Learning in a Global Environment, Berlin (Germany), May 2002.
- [2] J. Harward, J. A. del Alamo, V. S. Choudary, K. DeLong, J. L. Hardison, S. R. Lerman, *et al.*, "iLabs: A Scalable Architecture for Sharing Online Laboratories", presented at the International Conference on Engineering Education 2004, Gainesville, Florida, October 16-21, 2004.
- [3] J. L. Hardison, D. Zych, J. A. del Alamo, V. J. Harward, S. R. Lerman, S. M. Wang, *et al.*, "The Microelectronics WebLab 6.0 - An Implementation Using Web Services and the iLab Shared Architecture", presented at the International Conference on Engineering Education and Research 2005, Tainan, Taiwan, March 1-5, 2005.
- [4] G. Viedma, I. J. Dancy, K. H. Lundberg, "A Web-based linear-systems iLab," American Control Conference, 2005. Proceedings of the 2005, 8-10 June 2005, pp. 5139-5144 vol. 7.
- [5] J. L. Hardison, K. DeLong, P. H. Bailey, V. J. Harward, "Deploying Interactive Remote Labs Using the iLab Shared Architecture", Frontiers in Education (FIE) Conference, Saratoga Springs, New York, October 22-25, 2008.
- [6] V. J. Harward, J. A. del Alamo, S. R. Lerman, P. Bailey, J. Carpenter, K. DeLong, *et al.*, "The iLab Shared Architecture: A Web Services Infrastructure to Build Communities of Internet Accessible Laboratories", *Proceedings of the IEEE* Vol. 96, No. 6, June 2008.
- [7] A. Jiwaji, J. L. Hardison, K. P. Ayodele, S. S. Tickodri-Togboa, A. Mwambela, V. J. Harward, *et al.*, "Collaborative Development of Remote Electronics Laboratories: The ELVIS iLab", Proceedings of the American Society for Engineering Education (ASEE) Annual Conference and Exhibition, Austin, Texas, June 14-17, 2009.
- [8] "NI Educational Laboratory Virtual Instrumentation Suite (NI ELVIS)", <http://zone.ni.com/devzone/cda/tut/p/id/3711>.
- [9] J. A. del Alamo, V. Chang, L. Brooks, C. McClean, J. Hardison, G. Mishuris, *et al.*, "MIT Microelectronics WebLab", chapter in T. A. Fjeldy and M. Shur, eds., *Lab on the Web: Running Real Electronics Experiments via the Internet*, John Wiley & Sons - IEEE, 2003, pp. 49-87.
- [10] K. Amaratunga, and R. Sudarshan, "A Virtual Laboratory for Real-Time Monitoring of Civil Engineering Infrastructure", ICEE, Manchester, UK, 2002.
- [11] K. P. Ayodele, L. O. Kehinde, O. Jonah, O. Ilori, E. O. B. Ajayi, O. O. Osasona, "Development Of An Operational Amplifier Virtual Laboratory Based On ILab Architecture And NI Elvis", Proceedings of the American Society of Engineering Education (ASEE) Annual Conference and Exposition, Pittsburgh, Pennsylvania, June 2008.
- [12] S. Gikandi, Masters Thesis, ELVIS iLab: A Flexible Platform for Online Laboratory Experiments in Electrical Engineering, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology (May 2006).

- [13] B. Harrison, Masters Thesis, Expanding the Capabilities of the ELVIS iLab Using Component Switching, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology (May 2008).
- [14] A. Jiwaji, Masters Thesis, Modular Development of an Educational Remote Laboratory Platform for Electrical Engineering: the ELVIS iLab, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology (August 2008).
- [15] "NI Educational Laboratory Virtual Instrumentation Suite (NI ELVIS): Hardware User Manual", <http://www.ni.com/pdf/manuals/373363d.pdf>.

AUTHORS

H. Soumare is with the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge 02139 USA (e-mail: hsoumare@mit.edu)

R. Shroff is with the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge 02139 USA (e-mail: rshroff@mit.edu)

J. L. Hardison is with the Center for Educational Computing Initiatives, Massachusetts Institute of

Technology, Cambridge 02139 USA (e-mail: hardison@mit.edu)

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

V. J. Harward is with the Center for Educational Computing Initiatives, Massachusetts Institute of Technology, Cambridge 02139 USA (e-mail: jud@mit.edu)

P. H. Bailey is with the Center for Educational Computing Initiatives, Massachusetts Institute of Technology, Cambridge 02139 USA (e-mail: pbailey@mit.edu)

K. K. DeLong is with the Center for Educational Computing Initiatives, Massachusetts Institute of Technology, Cambridge 02139 USA (e-mail: kirky@mit.edu)

This article was modified from a presentation at the REV2009 conference in Bridgeport, CT, USA June 2009. Submitted May 29, 2009,. Published as resubmitted by the author(s) on July 20, 2009..