AC 2012-4622: REMOTE REALISTIC INTERFACE EXPERIMENTATION USING THE EMONA DATEX BOARD

Oluwapelumi Olufemi Aboluwarin, iLab Obafemi Awolowo University, Ile-Ife
Dr. Kayode Peter Ayodele
Prof. Lawrence O. Kehinde, Obafemi Awolowo University

Lawrence Kehinde has been a professor of electronic and electrical engineering at the Obafemi Awolowo University, Ile-Ife, Nigeria since 1988. He was the Director of ICT as well as the founding Principal Investigator of the University's iLab group in collaboration with MIT, USA. Currently, he coordinates a State Research and Educational Network. His present work includes developing virtual and remote labs for students' experimentation. He just concluded a three-year visiting Professor job at the Texas Southern University, Houston.

Mr. Babatunde Isaac Ishola

Remote Realistic Interface Experimentation Using the Emona DATEx Board

Abstract

Past work on remote experimentation with the Emona DATEx Board largely interacted with only one experiment already fixed by cable hardware at the remote end. This can be very restrictive and inflexible. In this paper, remote experimentation is being designed for the board in such a way that experiments can be altered and reconfigured by students through remote node to node cable connections.

Remote reconfiguration of the circuit for multiple experiments on the actual Emona DATEx is facilitated through the use of a National Instrument (NI) SCXI 1169 100-channel switch array and a web service implemented with C# that allows the switches in the array to be shared by multiple labs. Data acquisition is achieved through a NI USB-6215 DAQ card controlled by a C#.NET experiment engine working hand in hand with a custom-built LabVIEW DLL for each experiment. The client of the lab is an exact replica of the Emona DATEx board with all its controls (switches, knobs and connection nodes) activated programmatically such that they can be controlled remotely through a webpage. It is implemented using JavaFX 1.3.

In this paper, we have considered the iLab implementation of four basic telecoms experiment (Amplitude Modulation, AM Demodulation, DSBSC Modulation and DSBSC Demodulation) but we have used the Amplitude Modulation lab as a specific case study. It has great potential for students in Communications Engineering.

The lab is built on the iLab Shared architecture, which provides a framework for rapid remote-lab development and deployment, and uses a three-tiered architecture comprising Client, Service Broker and Lab Server. It is built on the NI Elvis System and Emona DATEx add-in module which is fully integrated with the NI ELVIS platform. The beauty of this work is that even though the student's interface is the replica of the Emona board, the art of actually making realistic and visible virtual cable connection remotely will add to improved pedagogy and understanding.

The technology and issues concerned with the development and usage are discussed. Furthermore, the experience gained in developing, testing and using the system, as well as feedback from students are also presented.

Keywords: Telecoms, JavaFX, Emona DATEx, iLabs, Elvis

I. Introduction

In the teaching of telecommunications engineering, practical experience substantially enhances the quality of the education. It substantiates a student's understanding of a subject and gives them the liberty to investigate systems that are difficult to model[1]. Unfortunately however, for a variety of logistic and economic reasons, conventional telecommunications courses in Nigerian educational institutions are often unable to include sufficient laboratory components to serve their students [2].

A possible remedy to this problem is the use of remote laboratories (rlabs). Rlabs are architectures in which users interact with remote real equipment using appropriate web browser-delivered user interfaces [3, 4]. They facilitate more efficient sharing of a wide range of lab resources like unique and expensive equipment and greatly simplify the logistic requirements of laboratory work like scheduling of equipment, lab space, staffing and safety of the users and equipment. Students can use a remote laboratory at any time and at any place as long as they are connected to the internet. Rlabs are therefore a viable means of combating the scarcity of laboratory experiments in telecoms.

It is common to use existing frameworks as bases when developing rlabs. Among other advantages, this leads to considerable reduction in development time[2]. The Massachusetts Institute of Technology (MIT) iLab Shared Architecture (ISA) is one of the most prominent of such frameworks. Rlabs based on it are simply termed iLabs [6,7,8,9]. The iLab platform provides a common framework on which laboratory equipment can be integrated to create remote laboratories and over the years, iLabs have been used to deploy a wide range of experiments cutting across various disciplines [7, 12, 14, 15, 16]. The iLab ISA creates a generic system for common tasks like authentication and data storage enabling the lab developer to focus more on tasks unique to the specific lab being developed.

In this paper, we describe the Amplitude Modulation (AM) iLab. The lab allows users to realize a basic AM circuit by remotely configuring an Emona Digital and Analog Telecommunications Experimenter (DATEx) training kit through mouse interaction with an interface. The lab attempts to make the user experience as close to that of the real lab as possible while ensuring that simplicity is not compromised.

The next section of this paper discusses the rationale behind the development of the lab, and provides an overview of design considerations. The next two sections are then devoted to explaining the hardware and software components used, detailing the interaction between them.

II. Rationale behind the Development of the AM iLab

The curriculum of the B.Sc. Electronic and Electrical Engineering program at Nigeria's Obafemi Awolowo University (OAU) contains 10 courses with subject matter targeting the communications sub-domain of the discipline. The curriculum also stipulates that between their third and final years, students take a total of three courses consisting entirely of complementary laboratory work.

Unfortunately, of the approximately 20 experiments that go into those practical sessions, only 5 are devoted to communications concepts. In addition, due to constraints on laboratory space and quantity of available equipment, students have to work in groups of between 5 and 8. Usually, less than half of the students in each group can be meaningfully engaged at any point in time.

Achieving an increase in the number of experiments has however been a big challenge for a number of reasons. The first is that the department has severe space constraints, sharing as it does a building with the Department of Physics of the same institution. With class sizes typically between 120 and 150, additional experiments would mean more space requirements to store the new hardware and accommodate more simultaneous sessions of student experiments. It would also imply more time and effort from the technical staff managing the labs. Considering the fact that the existing technical staff members are already overstretched, this in turn would require the recruitment of new lab staff. Finally, all of the above would require the expenditure of considerable sums of money, which the department does not possess.

Recently, the department received a donation of eight National Instrument (NI) Educational Laboratory Virtual Instrumentation Suite (ELVIS) experiment workstations from National Instruments. Coupled to low-cost computer systems obtained by the department, the ELVIS stations have been instrumental in the achievement of major improvements in the third-year laboratory work of students in the department. One advantage of using the ELVIS is that by changing the panel on top of the ELVIS, laboratory operators can employ the same ELVIS workstation for experiments spanning very large areas of electrical engineering and computer science.

One of the more popular ELVIS panels is the Emona DATEx, which is designed for telecommunications and signal processing courses and can be used for a broad range of introductory analog & digital modulation experiments. Naturally, the department considered obtaining Emona DATEx boards to expand the quantity of telecommunications experiments for its students. To do this however, the department would need to make considerable fiscal sacrifices elsewhere. Added to this was the very real worry that exposing the boards to more than 100 students per semester would surely lead rapidly to their being damaged.

Since rlabs address the problems noted above, it was decided that an rlab offering telecommunications experiments be developed around one or two Emona DATEx boards. The iLab architecture was chosen as bases because the university has done a lot of previous work using the platform. One of the most interesting aspects of iLabs to the university is the fact that the division of the framework into tiers makes it possible for institutions in bandwidth-constrained areas to use and share rlabs[2]. In particular, the iLab Batched Architecture has been used for a number of iLabs by the university.

The Batched architecture (Figure 1) is a three-tiered architecture consisting of a Lab Client, a Service Broker and a Lab Server [5]. The Client is an interface through which interaction between the user and the hardware takes place. The Service Broker handles issues regarding authentication and management of individual user accounts. The Lab Server is responsible for the interaction between the Lab Client and the experiment hardware. Its subcomponents include the Lab Server web service and Experiment Execution Engine, or Experiment Engine. Communication between the tiers takes place through web services using Simple Object Access Protocol (SOAP) as the communication protocol. The use of SOAP and web services makes the system platform independent [4].

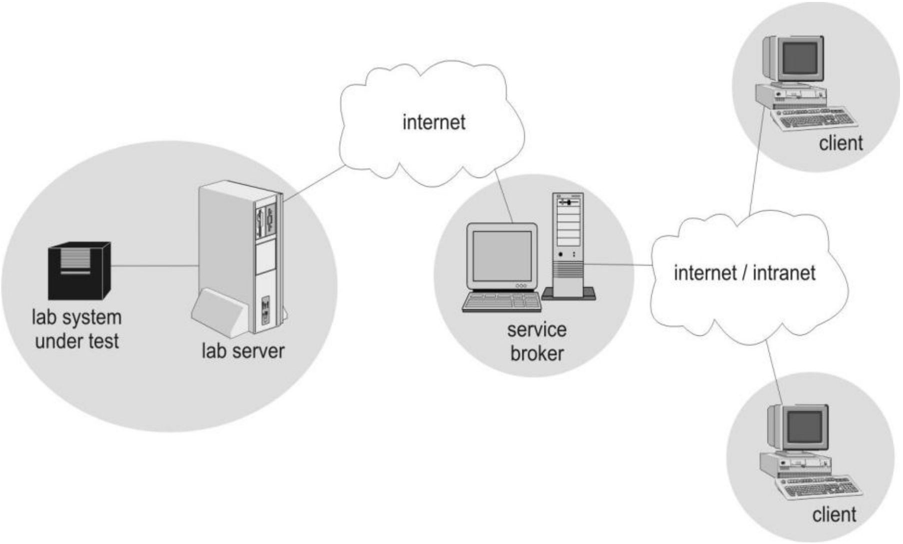


Figure 1: MIT iLab-Batched Architecture [10]

In developing the AM iLab, a number of requirements were identified. We wanted students to have an experience as close to a real laboratory session as possible. This played a key role in our choice of interface metaphor and choice of implementation software. A realistic interface was chosen to actualize our aims. A major implication of the desire to make the AM iLab telecommunications experiments similar to ones in a real lab was the fact that students needed to be able to modify the connections on the remote Emona board. This implied a need for a bank of remotely-controllable switches to allow system reconfiguration. Finally we wanted the lab to be available to students across a wide range of software and hardware platforms. This implied the use of a platform-independent basis for our client. While C# had been used in the past for the same reason[12], there was little doubt that use of Java would lead to clients that can be used on a larger number of systems.

There have been previous works on remote experimentation using Emona Datex. Notable ones implemented with the iLab architecture were developed in University of Makerere Uganda and Massachusetts Institute of Technology. Both labs are based on the Interactive iLab architecture that gives students real-time control over the lab set up one at a time. The interface the user interacted with was developed in LabVIEW. The interface displayed the block diagram for the Amplitude Modulation system implemented and permitted students to tune various controls and observe the behavior and changes of relevant signals. The limitations of the interface was that the user did not have a proper perspective of what he was dealing with based on the interface provided especially in terms of the hardware architecture which will affect the pedagogical value. The interface was also quite restrictive as the cable connections were already made for the users making usage relatively inflexible. In addition, Since development was done in LabVIEW, users need to have the LabVIEW runtime installed on their system before being able to view the lab interface on a web browser which is considered voluminous for bandwidth constrained regions like ours as most students do not have it installed, the LabVIEW runtime is considered large to be downloaded by students as it is close to 100 MB.

The resulting lab was built on an NI ELVIS equipped with an Emona DATEx trainer kit, and NI Switch Array. It has a JavaFX Client and a C# Experiment Engine. An overview of the AM iLab architecture is presented in Figure 2, and a more detailed treatment of the constituent parts will be undertaken in the next two sections.

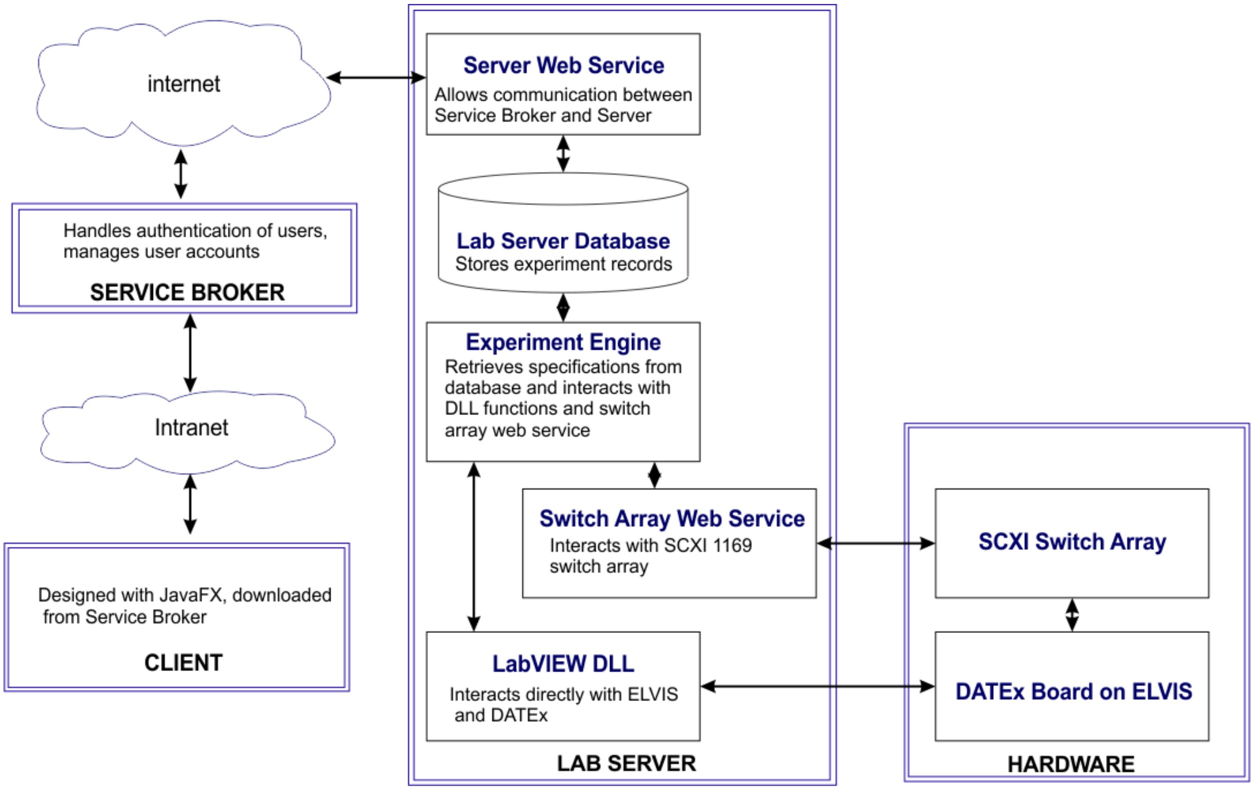

internet
Server Web Service
Allows communication between Service Broker and Server
Handles authentication of users, manages user accounts
SERVICE BROKER
Lab Server Database
Stores experiment records
Experiment Engine
Retrieves specifications from database and interacts with DLL functions and switch array web service
Intranet
Switch Array Web Service
Interacts with SCXI 1169 switch array
SCXI Switch Array
Designed with JavaFX, downloaded from Service Broker
CLIENT
LabVIEW DLL
Interacts directly with ELVIS and DATEx
DATEx Board on ELVIS
LAB SERVER
HARDWARE

Figure 2: AM iLab Architecture

III. Hardware Components

The hardware components in the AM iLab include an Emona DATEx Trainer Kit on an NI ELVIS, and an NI Switch Array.

a. NI ELVIS and Emona DATEx

The NI ELVIS was the hardware platform of choice because a couple of iLabs have already been developed based on it [12, 20]. It would therefore be relatively easier to develop a new iLab based on it. In addition, remote control of all NI ELVIS instruments can be achieved through Laboratory Virtual Instrumentation Experimentation Workbench (LabVIEW), an approach already used in the previous labs[12, 15].

The Emona DATEx ELVIS board has a collection of modules that can be combined to implement dozens of communications and telecommunications blocks [10]. Access to the modules of the DATEx board is made possible through an ample supply of terminals, knobs and switches as can be seen in its image in Figure 3. All DATEx knobs and switches can be varied either manually or under the control of LabVIEW Virtual Instruments (VI's)[10].

b. Switch Array

The modules on the DATEx can be interconnected in different ways, depending on the objective of an experiment. In a traditional laboratory, these connections are made using cables. In the AM iLab, an NI SCXI-1169 switch array is employed to achieve such flexibility.

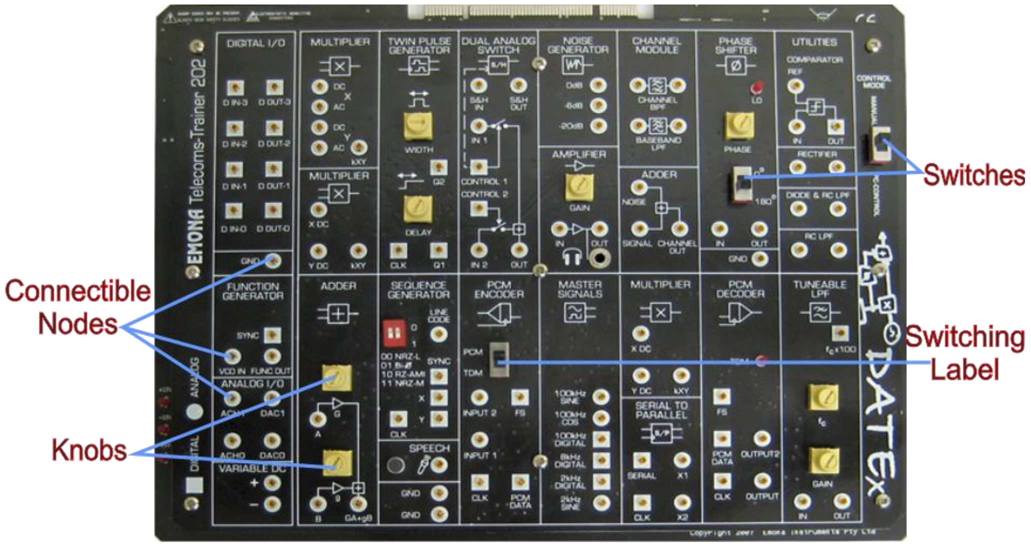


Figure 3: Emona DATEx Board showing all the available modules and important components labeled.

The 1169 is a 100-channel general-purpose switch module featuring SPST (Form A) armature relays. Ideal for medium-power automated test loads, this module offers high relay density, with 100 latching relays in a single SCXI slot. The NI SCXI-1169 can switch up to 100 V or up to 1 A. [11].

IV. Software components

The AM iLab has three software tiers. The Service Broker, which mediates between the Lab Server and the Lab Client, is an off-the-shelf software module whose operation has previously been described in the literature [1, 5, 7, 8]. Consequently, only the latter two tiers are discussed in this section.

a. Lab Client

The Lab Client is launched from within the Service Broker, which ensures that only authenticated users are granted access to it. It employs a realistic user interface [21] as shown in Figure 4. All controls, indicators and wire connections were implemented in a way that mimics real systems or existing software. A user that has used a real NI ELVIS or software like NI Multisim, would find it easy to use the AM iLab interface. This is in line with the Lazy User Theory [18, 19].

The Lab Client was developed with JavaFX, a platform for creating and delivering Rich Internet Applications (RIA) that can run across a wide variety of computer platforms [13].The graphics capability of JavaFX makes it easier to develop applications with graphical features especially when compared to languages like Java, C#, Microsoft Silverlight and LabVIEW, which have been used in developing iLabs [8, 12,14,15,16].

JavaFX is compiled to Java byte code, so JavaFX applications can run on any desktop and browser equipped with the Java Runtime Environment (JRE) [13]. Hence users with the Java runtime installed will not need to install additional software to run the lab client in the web browser.

As seen in Figure 4, the primary element in the Client interface is an image of a real DATEx board. To introduce an element of realism, students interact with the interface by “manipulating” the knobs and switches seen on the image. The various features incorporated into the design are described in the subsequent sections.

i. Knobs

The static knobs seen in the interface image were made dynamic by creating an invisible layer over the DATEx image. Programmed controls were then placed on this layer, with each control corresponding to a particular knob. Each control contained an image whose orientation could be programmatically changed to depict the knob it represented in different positions. Subsequently, mouse events were added to detect clicks and rotations which correspond to turning a real knob.

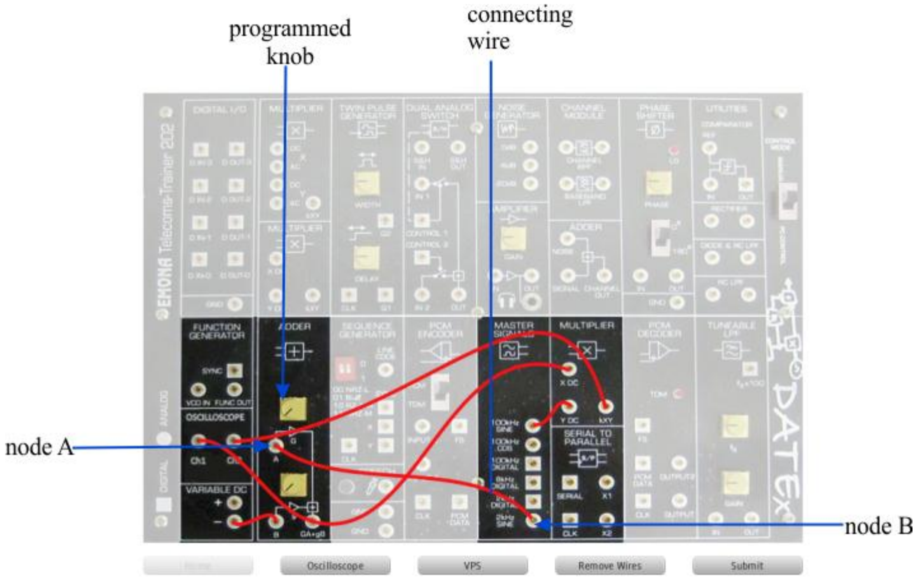


Figure 4: Lab Client Interface with wires connected and some components labelled for the AM iLab.

ii. Switches

All switches on the AM iLab Client were developed in a manner similar to that of the knobs. The programmed switches were placed on the invisible layer on top of the DATEx board image embedded in the Client interface. The model of development involved the use of two distinct images of the switch; one representing the switch in the “on” position, the other in the “off” position. When a mouse click event is detected, the images are toggled, to indicate the toggling of the corresponding switch between “on” and “off” positions.

iii. Wire Connection

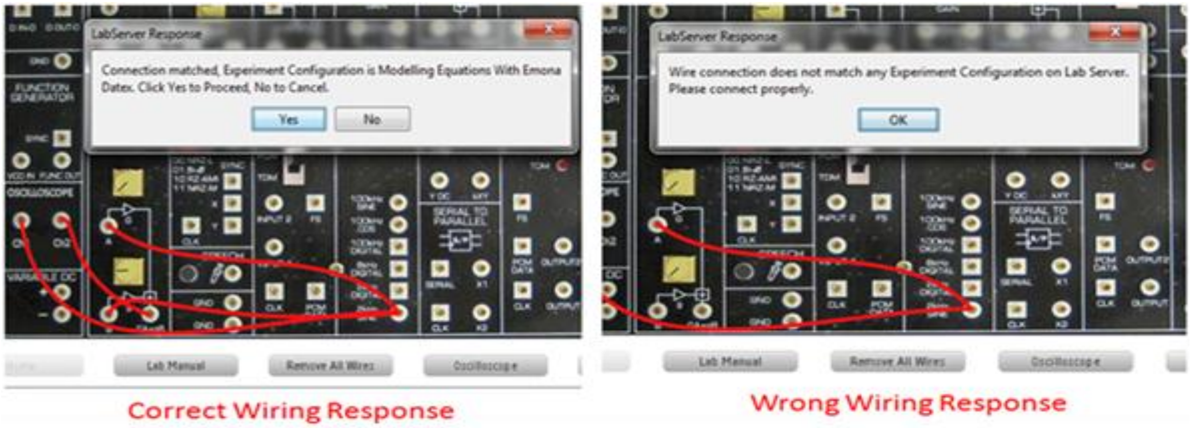
Wire connection involved mapping and identification coordinates on the board. Invisible labels were placed on every connectible node (some of which are indicated in Figure 3). Any mouse click within the area covered by the label is seen as a mouse click on the node. To reduce the need for the student to precisely position the mouse cursor over a node, the corresponding overlaying label was made to be larger than the node. Mouse events were attached to the labels to detect “click” and “drag” actions.

When the mouse cursor is pressed on a connectible node, dragged and then released over another connectible node that is within the set of valid connections for the lab, the image of a wire is inserted between the two nodes, indicating a connection of the two nodes. In Figure 4, the wire connection between the node labeled A and B appeared after the mouse cursor was pressed on Node A and dragged to and released over Node B.

iv. Additional Features

Students were given the liberty to make wire connections between any pairs of nodes they wished. This flexibility mirrors what students would experience in a real lab with a real DATEx board. Students could also delete connections between nodes simply by right-clicking on the "wire" image to be deleted, and then selecting the delete option from a pop up displayed afterwards.

Once the student has finished making his connections and manipulating the knobs and switches to the required positions, he can submit his experiment for execution. Before submission, the Lab Client verifies the connections and ensures it matches one of the configurations available on the Lab Server. If it does, the experiment specification is forwarded to the Lab Server in XML format through SOAP web services for the experiment to be executed. Figure 5 shows two possible results of the verification process: In Figure 5(a), the student's connections are correct, while Figure 5(b) shows the response to incorrect connections.



LabServer Response

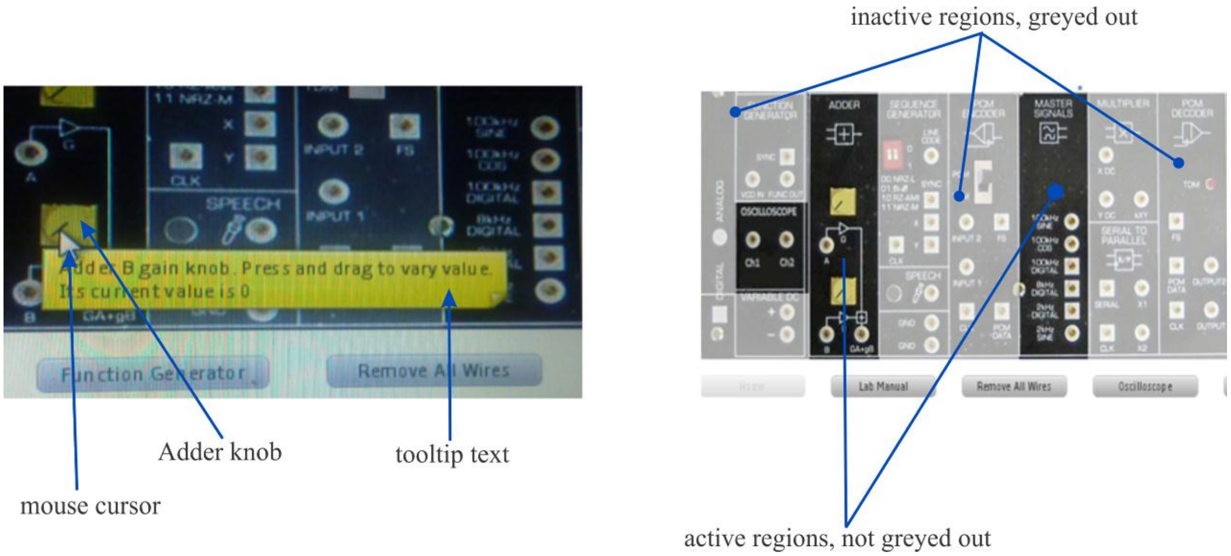
Connection matched, Experiment Configuration is Modelling Equations With Emona Datex. Click Yes to Proceed, No to Cancel.

LabServer Response

Wire connection does not match any Experiment Configuration on Lab Server. Please connect properly.

Figure 5: Lab Client response for correct and incorrect wiring

One downside to using a realistic interface is the risk clutter, which can lead to student disorientation. From previous studies, it has been observed that confusing user interfaces lead to frequent and/or serious errors in data handling [17]. To avoid this negative effect, modules that are not relevant to a specific experiment were greyed out so that the user would focus more on the active modules. Tooltip texts were attached to relevant components so that the user would have a good idea of the function of such components when the mouse cursor hovers over them. Figure 6 shows the tooltip text and the fade mechanism incorporated into the Lab Client.



Adder B gain knob. Press and drag to vary value. Its current value is 0

(a) (b)

Figure 6: Lab Client interface showing (a) the use of tooltip texts and (b) fade mechanism.

b. Lab Server

The DATEx Lab Server is the software component that handles the actual control of the hardware. This is achieved through the use of other subcomponents with specialized functions. Its subcomponents are the Experiment Engine, Lab Server database and the LabVIEW VI for hardware control.

Communication between the Service Broker and the Experiment Engine is achieved indirectly through shared access to the Lab Server database. When a user completes an experiment, experiment specification from Lab Client gets to Lab Server database through the Service Broker. When experiment execution is complete, the experiment result is dumped in the Lab Server database, after which the Service Broker is notified to pick up the result to be handed over to the Lab Client. The default database engine used by the iLab architecture is Microsoft SQL server 2005.

The Experiment Engine is the component responsible for retrieving the experiment specification (a document in XML format) from the Lab Server database, parses the XML, and retrieves unique parameters to run the experiment with. The Experiment Engine was developed in C# and it was chosen because it has good interaction with Microsoft SQL Server since they are from the same vendor.

The Experiment Engine achieves control of hardware through LabVIEW, which is well integrated into the ELVIS system and can easily send commands for hardware configuration. To make this possible, a custom LabVIEW VI was built to have full control over the hardware such that all parameters the experiment involves can be varied from it. This VI is then exported as a

Dynamic Linked Library (DLL) where all controls are reduced to parameters of a single method and a call on the method name is equivalent to running the custom VI when it is configured with the parameters within the method call. The DLL developed can be called from a text based programming language and receives input parameters to effect the desired configuration on the hardware and also returns a list of numbers representing the digitized form of the experiment such that the waveform can be re-plotted using those values.

V. The Lab Setup and Typical Usage Scenario

Although the lab can potentially support experiments on many types of amplitude modulation, only one is currently implemented: "ordinary AM", in which a DC offset is added to the message signal. In such a system, the modulated signal, X_{AM}, is given by:

$$X_{AM} = (X_M + X_{DC}) \times X_C$$

where X_M is a message signal, X_C is a carrier signal, and X_{DC} is a DC offset.

In the AM iLab, a 2kHz sine wave generated on the DATEx as one of its "master signals", is used as the message signal. The DC offset is a 6V DC supply, while the carrier is a 100 kHz sinusoid, also a part of the Master Signal module of the DATEx. The DATEx adder module is employed to sum the message and DC signals, and this sum is multiplied by the carrier signal using the multiplier module. The resulting output is the desired AM signal and its waveform can be viewed on the oscilloscope.

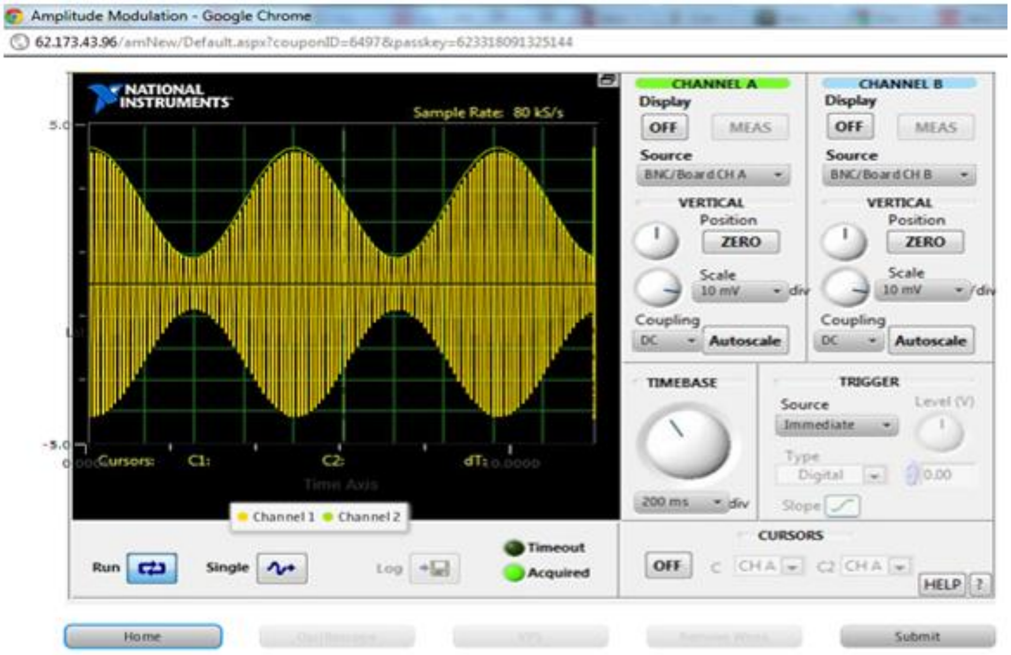
To work on the AM iLab, a student must log into the Service Broker, select the required user group, and launch the Lab Client window. Once the Client is launched, a typical experiment session starts with the student making wire connections between nodes. The student must draw wires connecting the two sinusoids (message and carrier signals), the adder module, and the multiplier module. In addition, a connection must be made between the output of the multiplier and the oscilloscope channel.

Once connections are done, the student must adjust knobs and switches to specify frequencies of the two sinusoidal signals. This is achieved by pressing and holding the mouse cursor which is immediately converted to a hand cursor when within range of executing the required action. The hand cursor can then be moved clockwise or anticlockwise to increase or decrease the inclination of the knob.

The user can switch views by clicking the "VPS" (for "variable power supply") button at the bottom of the lab client as seen in Figure 4. This directs the user to the VPS screen where the DC voltage can be adjusted. The VPS screen is a tweak of the default Elvis software VPS screen.

When all settings are completed, the student can submit the experiment for execution. The “submit” button is used to submit experiment specification to the lab server. The button stays inactive until the connection for the selected lab has been verified, after which the user has an option to proceed with submitting. Once the specification is verified to be valid, it is forwarded to the lab server for execution.

The “oscilloscope” button switches to the oscilloscope view displaying the result (waveform) of both channels of the oscilloscope. The volt-per-division and timebase values can be adjusted for the channels to help proper analysis of the oscilloscope output. The scope view for a sample experiment session can be seen in Figure 7. The message signal is in blue and the amplitude modulated signal is in green.



Amplitude Modulation - Google Chrome

62.173.43.96/amNew/Default.aspx?couponID=6497&passkey=623318091325144

Sample Rate: 80 kS/s

CHANNEL A: Source BNC/Board CH A; Scale 10 mV /div; Coupling DC

CHANNEL B: Source BNC/Board CH B; Scale 10 mV /div; Coupling DC

TIMEBASE 200 ms /div

TRIGGER: Source Immediate; Type Digital; Level (V) 0.00

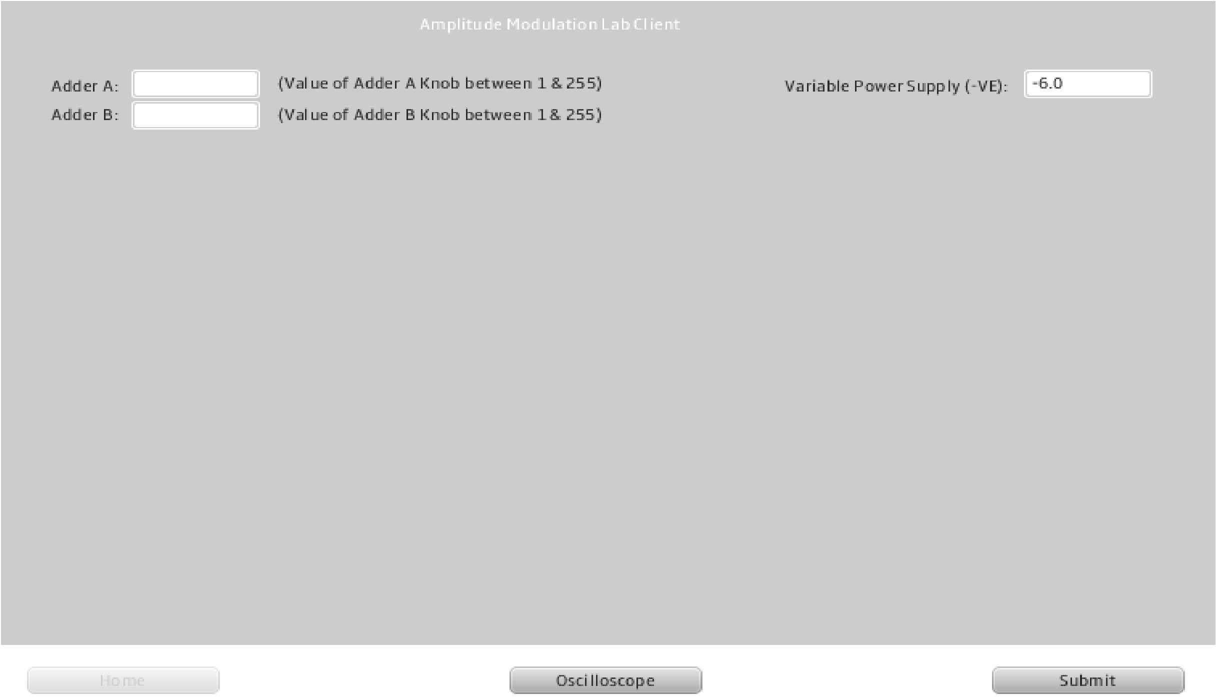
Figure 7: AM iLab Oscilloscope waveform.

VI. EVALUATION AND DISCUSSION

The AM iLab is a work in progress and has not been used in a credit-awarding course. However, some evaluation of some of the basic notions behind its development has been carried out. The primary concerns were whether the lab was useful for students, whether the use of a realistic interface made any difference, and ultimately whether the lab showed any promise in solving the problem of telecommunications experiments noted in section II.

A group of 15 student volunteers was used for the evaluation. Students were randomly selected from a class of approximately 150 students that just recently completed a 3-unit communications course in which they were introduced to modulation concepts.

All students were provided with laboratory manuals and asked to carry out the ordinary AM exercise on the AM iLab. However, while 8 of them were given access to the normal JavaFX interface, 7 were asked to use a new interface that simply provided text boxes into which they could enter values that corresponded to the amplitudes of the different signals in the modulation exercise (Figure 8). This was to offer some idea of the effectiveness (either real or apparent) of the lab.



Amplitude Modulation Lab Client

Adder A: (Value of Adder A Knob between 1 & 255)

Adder B: (Value of Adder B Knob between 1 & 255)

Variable Power Supply (-VE): -6.0

Figure 8: AM iLab Client Interface B

On completion, the students were asked to carry out a few tasks. First, each student was asked to draw the block diagram of the modulation scheme they had just used. They were given a 4-minute time limit to do this. In evaluating student's ability to draw the block diagrams, a pass-or-fail approach was used. In other words, if there were any major error in a student's block diagram, he was adjudged to have gotten the answer wrong. Subsequently, a short questionnaire was administered to the students. The questionnaire had 3 questions:

1. Do you feel you were interacting with real hardware?
2. Is the laboratory interesting or worthwhile?
3. Does this interface correctly reflect what you would have done in a real lab?

Table 1 shows the results of the post-experiment activities. The results in Table 1 suggests that having performed the AM iLab experiment using interface A, users had a better understanding of what amplitude modulation entails. The interface aided their understanding by walking them through the step wise process in creating an AM signal. This was made possible because every wire connection was an integral part of the AM block diagram and it helped the user visualize the components of the AM signal. Users therefore knew which signals should be added and multiplied to realize the AM signal due to the repetitive nature of making connections. It is also evident from the results in the table that those who used interface A were more likely to sketch the block diagram representation correctly. The use of interface A seems to have helped reinforce the notion to the student that the experiment they interacted with was real.

Table 1: Feedback from students after using interfaces A & B*

Evaluation Criterion	Students who used interface A		Students who used interface B	
	Yes	No	Yes	No
Did student accurately sketch block diagram?	5	3	2	5
Did student feel he was interacting with real hardware?	7	1	0	7
Did student feel the lab was interesting?	6	2	4	3
Did student think the lab accurately reflect a real lab?	7	1	0	7

**Interface "A" was the realistic JavaFX interface while interface "B" was the blank one shown in Figure 8.*

Interface B on the other hand did not involve users in making wire connections hence they were left to assume the state of things at the backend. The users could not go through the process of visualizing the constituent components of an AM signal therefore the interface was not really of help in making them sketch the block diagram correctly. User remarks concerning interface B also made it evident it was not considered worthwhile or enjoyable, especially when compared to interface A.

Obviously, not too much meaning can be attached to the evaluation above, as it uses a very small sample size, and experiment could certainly have been better-designed to yield more conclusive results. At the very least however, the results above provide reason for optimism that the ideas behind the design of the AM iLab have some basis to them.

VII. Conclusion And Future Work

The AM lab showcases how students can be exposed to the huge benefits of the DATEx alongside NI Elvis in a cost-effective way using iLabs. The conventional method of conducting telecoms experiments in the department employs the use of modules for different telecommunications experiment i.e. a single one for AM, FM and the likes. The experimental setup described helps to largely cut down cost in creating telecoms experiments by giving users infinite and unrestricted access to the lab setup. The hardware utilized allows for the implementation of multiple telecoms experiment thereby conserving lab space and largely cutting down cost. The lab interface also helped give students a feeling of having real access to the lab equipment thereby largely improving the user experience and helping students learn what is required thereby actualizing the aim of the lab instructor.

The AM Lab which is the subject of this research paper describes one out of five experiments deployed on a single DATEx board. The switch array is used to switch between different experiments configurations. The process described here is the same for the other labs the major difference is the connection of wires and the DATEx modules used.

Future work involves increasing the number of experiments available on the client interface so as to cater for other concepts in telecoms engineering. In addition, since the Lab Client was developed with JavaFX 1.3, the present code base will be ported to JavaFX 2.0 which is an overhauled version of JavaFX that makes all the JavaFX libraries available within the JavaSE environment. This will lead to better stability of the system and faster loading time.

Acknowledgments

The authors would like to thank members of the Centre for Educational Computing Initiatives (CECI) at the Massachusetts Institute of Technology for numerous helpful suggestions. This research was funded by grants from the Carnegie Corporation of New York. National Instruments kindly donated some of the equipment used for the research and we will also like to thank Andrew Watchorn of National instrument for providing numerous hints on advanced LabVIEW programming.

References

1. J. L. Hardison, K. DeLong, V. J. Harward, J. A. del Alamo, R. Shroff, and O. Oyabode, "Enabling Remote Design and Troubleshooting Experiments Using the iLab Shared Architecture", 2010.
2. L. Kehinde, X. Chen, K. Ayodele and O. Akinwale, "Developing Remote Labs for Challenged Educational Environments", chapter in: *Internet Accessible Remote Laboratories: Scalable E-Learning Tools for Engineering and Science Disciplines*, IGI Global, Edited by Abul K. M. Azad, Michael E. Auer and V. Judson Harward.
3. I. Gustavsson, J. Zackrisson, H. Akesson, L. Hakansson, I. Claesson.and T. Lago, "Remote Operation and Control of Traditional Laboratory Equipment", Vol. 2, Issue 1, pp.1-8, 2006.
4. G. Viedma, I. J. Dancy, and K. H. Lundberg, "A Web-based Linear-Systems iLab." American Control Conference, Portland, OR, Jun.8–10, 2005.
5. V.J. Harward, J.A. del Alamo, V.S. Choudhary, K. deLong, J.L. Hardison, S.R. Lerman, J. Northridge, D. Talavera, C. Varadharajan, S. Wang, K. Yehia, D. Zych., " iLab: A scalable architecture for sharing online experiments.", Gainesville, Fl, October 2004. ICEE.
6. O.B. Akinwale, K.P. Ayodele, A.M. Jubril, L.O. Kehinde, O. Osasona, O. Akinwunmi, A.T. Asiimwe, C. Mwikirize, P.I. Musasizi, S.T. Togboa, A. Katumba, J.Butime, J.P. Nombo, M.M. Baraka, S. Teyana, M.J. Alfred, K.M. Musa, "Online Laboratories: Enhancing the Quality of Higher Education in Africa", in Proc., Conference of Rectors, Vice-Chancellors & Presidents of African Universities (COREViP), South Africa, 2011 .
7. 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., International Conference on Engineering Education, Valencia, 2003.
8. J. 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," in Proceedings Of The Conference On Exploring Innovation In Education And Research, Tainan, Taiwan, 2005.
9. V.J. Harward, J.A. del Alamo, S.R. Lerman, P.H. Bailey, J. Carpenter, K. DeLong, C. Felknor, J. Hardison, B. Harrison, I. Jabbour, P.D. Long, T. Mao, L. Naamani, J. Northridge, M. Schulz, D. Talavera, C. Varadharajan, S. Wang, K. Yehia, R. Zbib, and D Zych, " 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

10. Barry Duncan, EmonaDatex Lab Manual for NIELVIS volume 1. Emona Instruments Pty Ltd, 2009.
11. http://sine.ni.com/nips/cds/view/p/lang/en/nid/10676, 2006, assessed 2011.
12. 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," in ASEE Annual Conference and Exposition, Pittsburgh, PA, 2008, pp. AC 2008-1098.
13. http://en.wikipedia.org/wiki/Javafx, 2011.
14. Olawale B. Akinwale, "Development of a Robust iLab Platform for Robotic Arm Experimentation," Dept of Electronic & Electrical Engineering, Obafemi Awolowo University, Ile-Ife, MSc Thesis 2010.
15. C. Mwikirize ,A.T. Asiimwe, L. Musasizi, V. Namuswa, M. Dawn Nakasozi, C. Mugga, A. Katumba, S.S. Tickodri-Togboa, J. Butime, P.I. Musasizi., "Development of Online Laboratories for Modulation and Combinational Logic Circuit Analysis Using NI ELVIS II," in 2010 Seventh International Conference on Information Technology, Nevada, Las Vegas, USA, 2010, pp. 1069-1073.
16. Kayode P. Ayodele, OlawaleAkinwale, Lawrence Kehinde, Oladipo O. Osasona, E.O.B. Ajayi, & O.O. Akinwunmi, "Advanced Digital Laboratory: An FPGA-Based Remote Laboratory for Teaching Digital Electronics," in Proc., ASEE Annual Conference & Exposition, vol. Paper AC 2009-1206, Austin ,Tx, 2009.
17. S.L. Smith and J.N. Mosier, "Guidelines For Designing User Interface Software", The MITRE Corporation Bedford, Massachusetts, USA, 1986.
18. M. Collan and F. Tetard, "Lazy User Theory Of Solution Selection", in International Conference on Cognition and Exploratory Learning in Digital Age (CELDA), 2007
19. Zipf, G. K. (1949). Human Behavior and the Principle of Least Effort, Addison-Wesley, Reading MA, USA.
20. Jiwaji, A., Hardison, J., Ayodele, K.P., Tickodri-Togboa, S.S., Mwambela,A., Harward, V.J., del Alamo, J.A., Harrison,B., and Gikandi,S. , “Collaborative Development of Remote Electronics Laboratories”: The Elvis Ilab.*Proceedings of the ASEE 2009 Annual Conference and Exposition*, June 2009, Austin, Texas.
21. Olowokere, D. O., Ayodele, K. P., Kehinde, L. O., Jonah, O. P., Akinwunmi, O. A., and Ajayi, E. O. (2008a). Realistic Looking Interfaces: In Search Of The Best Ergonomic Metaphors For Remote And Virtual Laboratory Interfaces. *Proc. of the 2008 ASEE Annual Conference & Exposition*. Paper AC 2008-1316. Pittsburgh, PA: ASEE.