

A Modular and Extensible Remote Electronic Laboratory

A. Bagnasco, G. Parodi, D. Ponta, A. M. Scapolla

Department of Biophysical and Electronic Engineering, University of Genoa, Genoa, Italy

Abstract — The remote control of instrumentation is not enough to set up a remote laboratory. Several pedagogical, technical, and structural issues must be faced to obtain modular and scalable systems. This paper reports our experience in developing a remote laboratory system to teach electronics to information engineering students. Our target is to develop proper architectures and tools, to obtain an easy management of the system and a better pedagogical effectiveness. The approach used in describing the components of online experiments facilitates the upgrade of the laboratory and the sharing of experiments, defined as standard learning objects. A particular attention has been paid to pedagogical issues. The main problems being investigated are: the lack of the support that students used to have in traditional labs, the reduced interaction with the objects under test, and the necessity of providing feedback to the experimenters.

Index Terms — electronic experiments, metadata for online experiments, remote laboratory

I. INTRODUCTION

Executing laboratory experiments via Internet is of great interest for the emerging e-learning scenarios: scientific curricula, where testing theories through practice is of primary importance, continuous education, generally constrained by time and space availability, and specialized education/training based on the use of expensive, rare and geographically scattered equipment.

Today's market offers tools and solutions for remote control of laboratory apparatus. Unfortunately, a remote laboratory is not the bare exchange of data between a web browser and the instrumentation, but it is characterized by a large set of issues related to pedagogical, technical, and organizational aspects, which are not so trivial to solve.

Multidisciplinary skills are required: teachers define the educational objectives and the best path for learning, laboratory technicians make the hardware set up available, and experts in computer-based instrumentation control and distributed computing develop the necessary software interfaces. We need the contribution of all these actors whenever the laboratory must be updated, to include new equipments and to add experiments.

The process is costly and time consuming if not adequately supported by tools that guarantee scalability, reliability and security.

This paper contains a brief overview of the remote laboratory ISILab (Internet Shared Instrumentation Laboratory) [1], which is currently used to deliver remote access to experiments on electronics for the benefit of some engineering courses.

Section 3 deals with the laboratory management issues and presents the tools used to handle both the hardware and the software parts.

Section 4 focuses on the description of the online experiment metadata schema used by our system, which aims at valorising on line experiments as learning objects.

Finally, section 5 investigates pedagogical issues and the solutions adopted to face the lack of support that students used to have in traditional labs, the reduced possibility of interaction with the real test bench, and the necessity of providing feedback to the experimenters.

II. THE REMOTE LABORATORY ON ELECTRONICS: ISILAB

ISILab is a remote laboratory for practicing on electronic instruments and measurement methods executing real experiments of scalable complexity on analog and digital circuits. The experiments deal with basic electronic measurements, such as the delays in digital circuits or the gain and the distortion of amplifiers, and use a waveform generator and an oscilloscope. The laboratory integrates experiences, lectures, exercises, and handbooks.

Users access the laboratory from a web portal, a unique access point that links several real laboratories via the Internet. Real laboratories can be distributed over a wide geographic area and are accessed seamlessly by users, whose activity is not influenced by the physical location of the experimental set up. The portal offers an indexing service of the available experiments, is responsible of the security policies, and initializes the direct communication between the client and the Real Laboratory Server (RLS) in charge of the selected experiment. The RLSs are connected to the instrumentation (IEEE 488, serial lines, PCI bus, etc) and control the experimental setups.

Each RLS acts as a scheduler of measurements; it can serve multiple users at the same time and guarantees the coherency of each experimental session; no booking facility is provided.

Every time a user starts an experiment, the RLS initializes a data space that contains the setting of the devices involved in the experiment. After this initialization, the engine waits for the client's requests, receives commands and queue them waiting until a free time slot becomes available.

A key-point of our approach is that users do not require any specific hardware or software to access experiments, except for a web browser and the Java Virtual Machine. Users execute the experiments interacting with appropri-

ate graphical interfaces, which allow to control the instruments and to move the probes on the circuits.

Switch matrixes let to share the same instrumentation among different circuits present on the same workbench. The selected circuit is dynamically connected to the instruments when the user demands it, and it stays connected just the time necessary to the measurement. Thus a workbench serves many users at the same time, using a time sharing technique. Users operate as if they had exclusive control of the experimental set-up.

Each experiment is presented by a detailed description, the electric diagram and the synoptic view of the workbench that shows the components and their connections. The documents, associated to each experiment, cover:

- the theoretical concepts that are behind the experiment;
- the goal and the expected results;
- the description of the experimental set-up,
- the proposal of a set of exercises;
- bibliographic references;
- handbooks of the instruments.

Real-time images of the experimental set-up are acquired by a web cam.

Users can execute the experiments in two different ways, called “guided” and “independent” mode. In guided mode there is a privileged user who is the only one able to modify interactively the operational conditions, acting on the instrumentation controls, and the other users are only able to see the response of the system on their computer screens. This mode can be very effective in a context of distance education as teachers can show real laboratory experiments via Internet. When an experiment is carried out in independent mode, all users are able to interact with instruments in parallel mode and see only the results related to their own commands.

Fig. 1 shows the execution of an experiment on the client side. The circuit under test is an integrator; it is based on an operational amplifier and the user is required to verify the circuit response when it is stimulated using different waveforms. The web page shows the electrical diagram of the circuit, and states the steps for the execution of the experiment. On the right, the virtual instrument control panels are visible.

III. ISILAB COMPONENTS AND MANAGEMENT TOOLS

In ISILab, both the lab web portal and the RLSs operate on the base of well structured configuration files. The lab web portal engine does not need to be modified to link a new RLS and to enlarge the number of workbenches, the RLS engine must not be changed to add new experiments to an existing workbench.

We can state that from the architectural point of view, the laboratory has a modular and scalable structure, but other components play relevant roles in the laboratory management. Anytime a teacher decides to deliver a new experiment, the remote laboratory requires defining the educational contents, setting up the hardware benchmark, and developing software interfaces to the equipments and the experiment itself.

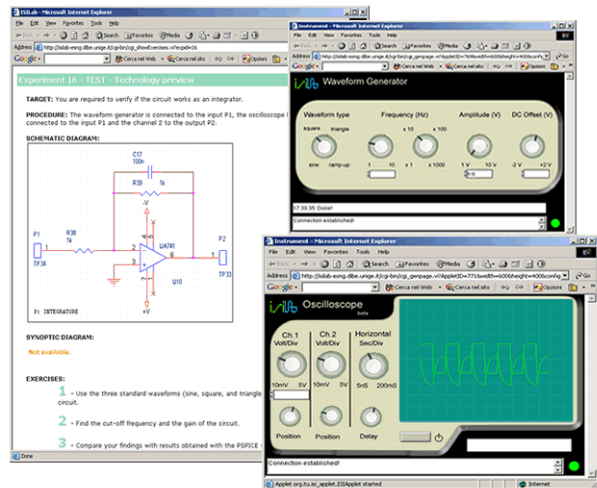


Figure 1: User interfaces

The process is expensive and time consuming if not adequately supported, thus we created proper tools to handle the elementary bricks (hardware devices, virtual interfaces, circuits under test, learning contents) and to assemble them easily.

A. Instruments

ISILab supports both stand-alone (rack-and-stack) instruments, as well as computer-based ones. It abstracts hardware devices (instruments connected via local buses, DAQ board, etc.) as resources that offer homogeneous APIs (Application Program Interfaces) to the RLS engine, a device-independent software layer, in charge of managing the communication with remote clients. The insertion of a new device does not require recompiling or modifying the RLS. It is sufficient to wrap the instrument driver with an appropriate driver adapter that exposes the device functionalities, and to change some configuration files. The driver adapter is a small software layer in charge of intercepting the calls from the RLS engine, translating and forwarding them to instrument drivers in the right format. It provides the connection between the virtual panel and the instrument driver and can embed mathematical functions or algorithms that extend the features of the device. For instance, an oscilloscope can be managed by a device driver that, executing the FFT (Fast Fourier Transform), allows using it as a spectrum analyzer. We have developed driver adapters for the instruments most common in an electronic measurement laboratory: oscilloscope, function generator, and digital multimeter. These are based on IVI [2] technology, in order to reuse them with instrument from different vendors, but belonging to the same class.

B. The Circuits under Test

Ideally, the best way to allow users to practice with different circuits is to let them compose the circuits starting from discrete components. This approach allows creating any circuit they want, with the only constraints of the available components. Moreover, the use of a virtual breadboard reproduces the task of assembling the circuit

exactly as it goes on in the real world. Remote laboratories adopting this approach are described in [3,4]. Unfortunately, the number of switches, required to assemble the circuits, increases exponentially with the number of the components. In case the circuits under test are not limited to very simple ones, this makes the approach very expensive; furthermore, as the signal must pass through several switches, it can become corrupted. Thus, the approach is suitable only when we deal with a reduced number of components. Another drawback is the possibility of creating dangerous connections and consequently, the need of validating the circuit design before applying the configuration to the switch matrix.

Such considerations suggested adopting a different approach. We have created a modular system named ISIBoard. It consists of a motherboard with sixteen slots, where we can insert cards hosting the circuits to test (see Fig. 2). Each card has an area of about 45 cm² where it is possible to build the circuit. Eighteen lines are available for providing the necessary power supply, and for connections with the instruments. In particular we have:

- five lines for power supply (+12V, -12V, power supply GND, +5V, signal GND)
- one line for input signals
- three lines for output signals
- eight lines for circuit identification.

When the circuit card is mounted on the motherboard, the connections between these lines and the instruments (power supplies, waveform generator, oscilloscope, etc.) are dynamically created by a set of switches properly controlled by the RLS. The circuit identification lines give us the possibility of associating an identifier bit code to each circuit. The goal is the automatic configuration of the hardware and the software environment on the base of the circuit description files that have been previously stored in the system.

ISILab, since from its early development stages [5], describes experiments and instrument panels in XML format. Now we are able to associate these sheets with the active circuits using a microcontroller that reads such identifiers and communicates them to the RLS. This approach is similar to the virtual TEDS that is an extension of the IEEE 1451.4 standard [6]. In this way it is possible to have plug-and-play facilities that allow non-technical persons mounting the circuits on the board.



Figure 2: The motherboard and the circuit cards

Using the ISIBoard, teachers can design the circuits according to their educational objectives, without limits on the number of components and, once these are assembled, each circuit occupies a slot in the motherboard. This allows proposing more complex circuits to the evaluation of the students without using complex switching systems. The reduced number of switches results in signals of better quality and in a cheaper laboratory setup.

C. Virtual panels

The instrument virtual panels are the graphical user interfaces (GUI) delivered to remote clients to interact with real devices. They contain graphical objects, such as knobs, menus, waveform charts. Each object has graphical properties and is bound to specific run-time parameters.

Virtual panels are defined according to a specific XML schema and the authoring of new panels is supported by the tool that is described in the next section.

A device-independent software module visualizes the GUIs and communicates with the core engine. A device can be controlled by different GUIs according to the target users' skills and the experiment requirements. Virtual panels' management does not impact on the core laboratory engine.

1) The ISIApplet and the Authoring Tool

Virtual panels allow presenting instruments customized according to the educational objective and to the target user profile. A simple panel, containing only the controls that are strictly necessary to the execution of the specific experiment, is more appropriate for a novice, than a complex panel offering a wide range of control possibilities. On a simple panel, the novice can focus the attention on the measurement and postpone the knowledge of the real instrument. Nevertheless, some situations require the most realistic, and thus complex, interfaces. If this is the case, the virtual panel development may become quite difficult and a considerable programming effort is required. This argument strongly impacts with laboratory expansibility and represents the main rationale behind our engagement in the development of a tool that simplifies the creation of instrument interfaces and allows assembling virtual panels with arbitrary functionality and look&feel. The basic idea is to have a general purpose, reconfigurable software module that can be used to control different instruments. Such application must be able to change its appearance and its behaviour according to a given configuration file. The creation of this file can be demanded to a user-friendly authoring tool.

This scheme is not new [7, 8 and 9], and some commercial products exploit it. For instance, Nacimiento [10] shipped a product called AppletVIEW. It offers a library of general control components, as knobs, switches, etc. Developers can assemble them using a graphical editor, and build a specialized Java GUIs. The editor application produces an output file that defines the graphical properties of the components of the virtual interface (type, position, dimensions, colour, etc.) and drives the behaviour of the applet used to instance how many virtual panels as required by the experiment.

AppletVIEW configuration files are written using an XML-based language called VIML (Virtual Instrument Mark-up Language). VIML merely describes the graphical objects and the associated parameters in a 1:1 ratio, and, in other words, any action on the virtual panels causes a transmission to the server. This simplistic approach limits the development of complex and realistic interfaces. Improvements in flexibility and data-exchange performance have been the main motivations to develop of a new tool for authoring instrumentation's interfaces. Our main goal was to override the limit of the 1:1 mapping between object manipulation and data transmission. Examples of actions that don't require data transmission are: pressing a shift key to change the operational mode of another key, changing the offset in waveform visualization.

The authoring tool captures different attributes for each graphical object: the cosmetic aspect, such as shape and colour; the functionality and the changes generated by the users' actions.

These attributes can be defined using a user-friendly editor and registered in XML format by the authoring tool [11].

The viewer is a Java applet that works on the base of these XML configuration file that are uploaded by the client together with the applet itself.

IV. ON LINE EXPERIMENTS METADATA

ISILab defines an experiment in terms of attributes related to the classification of the experiment as a learning object, instruments and virtual interfaces used to conduct the experiment, and assignments that students are requested to carry out.

Fig. 3 maps the schema of the experiment data structure. It is very similar to the IEEE LOM standard; we propose some extensions in order to treat the specificity of these learning objects and to store all the information that allow the ISILab engine to work properly.

The green line groups round the metadata compatible with the LOM standard, the blue line collects the data that are specific to ISILab.

A remote lab experiment can be thought as a sequence of measurements triggered by the actions of the remote clients on the virtual panels. In consequence of each action, the RLS engine evaluates which changes must be activated on the real instruments. We need to translate actions into ordered sequences of calls to the driver adapters of the instruments involved in the measurement. Each call passes parameters and receives a feedback.

The RLS engine acts as a scheduler on the base of the experiment description file. The order used to list the instruments drives the schedule of the calls to the instruments.

Once more adding an experiment to the laboratory does not change anything in the core engine and can be accomplished by the creation of the XML file that describes the experiment.

V. PEDAGOGICAL ISSUES

From an educational point of view replacing traditional with remote laboratories is not only a matter of developing proper technical tools but requires an innovative educational approach and a close interaction between technical and pedagogical aspects. A good understanding of the pedagogical aspects is essential for an efficient use of remote labs.

The early experiments carried out by our team were targeted to students of the first year of information engineering. They consisted of the measurement of electrical parameters of elementary digital and analog circuits. Given the large number of attending students and the close monitoring activities, the experiments have provided a large amount of data to drive the technical development of ISILab and its pedagogical use. The main problems being investigated are: the lack of the support that students used to have in traditional labs, the reduced interaction with the objects under test, and the necessity of providing feedback to the experimenters.

At the first year, the students' technical background is so limited and diverse that we cannot count on any familiarity with measuring instruments and circuit breadboards. Thus, we decided to run a set of activities that we call "Prelab" before each experiment. They consist of tutorials to train students on the use of instrumentation and to provide the background necessary for understanding each experiment. The access to the experiment is allowed only upon a positive individual test on the issues explained in the tutorials. Such pedagogical actions are served by a Learning Management System. In our case, the choice has been the open source LMS Moodle.

As explained in section 3, ISILab does not allow students to assemble their own circuits, but the possibility of connecting the oscilloscope probes to several different test points in the circuit provides a more realistic testing environment.

A problem we found in the past experimentation phase has been the difficulty of verifying the correct execution of the experiments and, therefore, of providing a feedback to students. A preliminary solution we are testing now is the delivery, after the experiment, of a self-assessment on-line questionnaire that should increase the awareness of the experimenter regarding the actions performed.

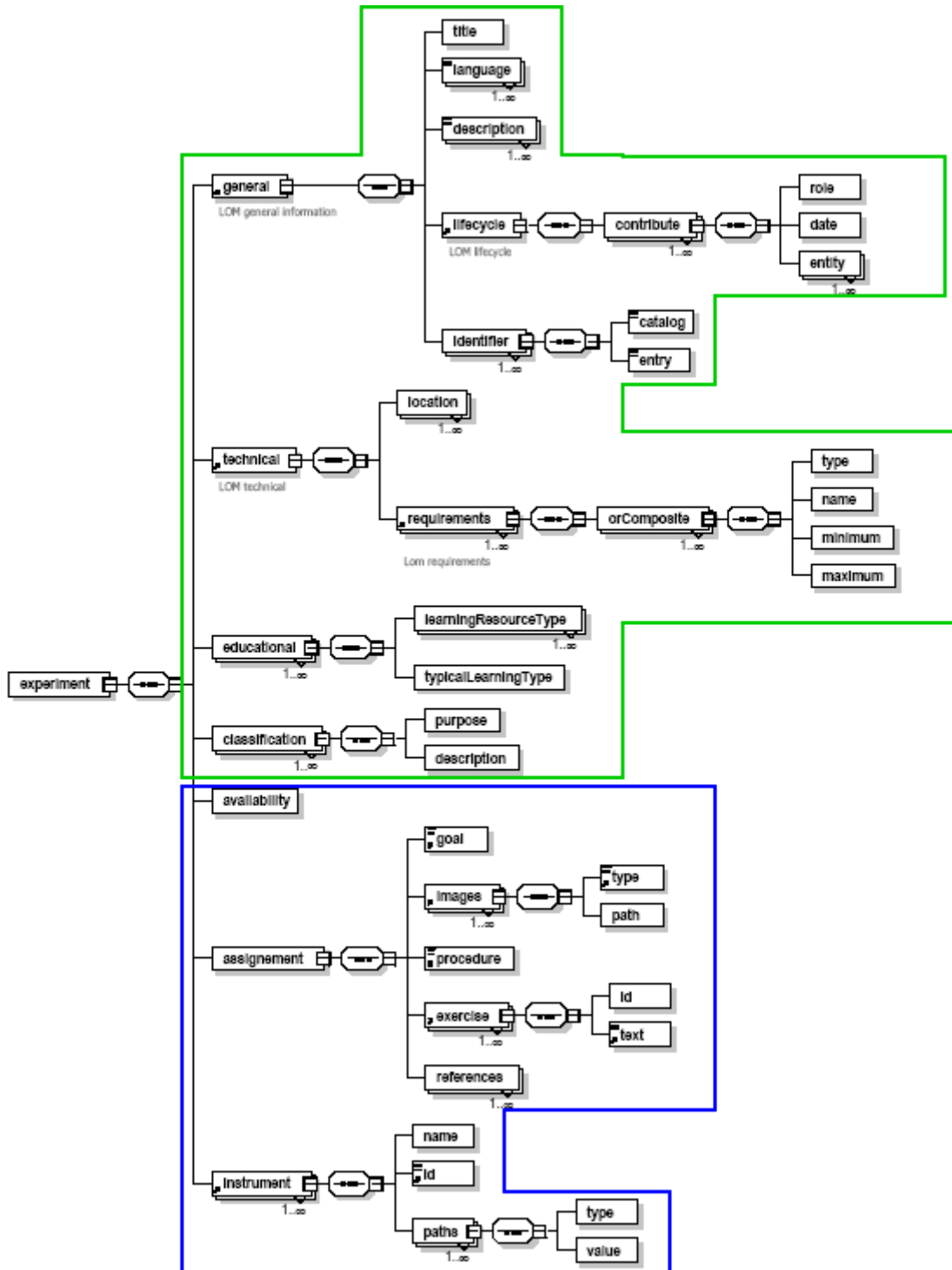


Figure 3: View of the experiment schema

VI. CONCLUSIONS AND FUTURE DIRECTIONS

Even if remote labs cannot replicate as a whole the experience of a traditional lab and do not develop exactly the same skills, for instance, the phase of construction of the prototype cannot be replicated efficiently in a remote environment and therefore the related skills cannot be developed at the same level, it must be considered that the evolution of hardware technology makes less and less realistic the experience of building from scratch a state-

of-the-art electronic system in the laboratory. The kind of learning associated with remote labs, we believe, is certainly in touch with current and, even more, future technologies.

The approach that we have adopted in managing resources (circuits, instrumentation, virtual panels) match the most important initiatives in the field of standards for e-learning [12] and makes easier both the improvements

to the online laboratory on electronics and our contribution to sharing experiments among a wider community.

We hope that our experience can contribute to stimulate a profitable collaboration among the developers of online laboratories.

ACKNOWLEDGMENT

The authors thank Giorgio Carlini for the precious contribution to the development of the ISIBoard system.

REFERENCES

- [1] A. Bagnasco, M. Chirico, G. Parodi, A. Sappia, A.M. Scapolla, "A Virtual Laboratory for Remote Electronic Engineering Education", in *International Perspective on Tele-education and Tele-learning*, Ashgate Book, 2000, pp. 1-14
- [2] <http://www.ivifoundation.org>
- [3] I. Gustavsson, H. Åkesson, "A Remote Laboratory Providing Teacher-defined Sessions", *Proceedings of the ICEE 2004 Conference in Gainesville, USA, October 17 - 21, 2004*.
- [4] Asumandu J. A. Et al., "A Web-Based Electrical and Electronics Remote Wiring and Measurement Laboratory (RwmLAB) Instrument", *IEEE Transactions on Instrumentation and Measurement*, Vol. 54, No. 1, February 2005
- [5] A. Bagnasco, M. Chirico, A. M. Scapolla, "XML Technologies to Design Didactical Distributed Measurement Laboratories", *IEEE IMTC2002, Anchorage, Alaska, USA*
- [6] <http://zone.ni.com/devzone/conceptd.nsf/webmain/F606F87ED4E9454786256EAF006DF5E0#1>
- [7] H. Saliyah-Hassane, P. Dumond-Burnett, C. Loizeau, "Design of a Web-Based Virtual Laboratory Instrument Measurement Interface", *International Conference of Engineering Education*, 2001 Oslo, Norway
- [8] C.C. Ko, Ben M. Chen, C. M. Loke, C. D. Cheng, X. Xiang, A. K. Eapen and T. S. Lim, "Automation in creating Java 3D-based Virtual Instruments", *International Conference on Software Engineering Research and Practice, Las Vegas, USA 2002*
- [9] <http://pioneer.gsfc.nasa.gov/public/iml/>
- [10] <http://www.nacimientto.com>
- [11] A. Bagnasco., A.M Scapolla, T. Vassileva, V. Vasilev, I. Furdziev, "Authoring and Automatic Generation of Virtual Panel For Controlling Instrumentation in a Remote Laboratory Context", *2nd International Conference on Multimedia ICT's in Education - m-ICTE 2003, Badajoz, Spain, December 3-6, 2003 - ISBN 84-96212-09-2*
- [12] <http://www.prolearn-project.org>

AUTHORS

A. Bagnasco, G. Parodi, D. Ponta, A. M. Scapolla are with the Department of Biophysical and Electronic Engineering of the University of Genoa, Italy. (e-mail: bagnasco, parodi, scapolla, ponta@dibe.unige.it).

Manuscript received May 25, 2005.