e-lab Remote Laboratory

Integrated Overview

http://dx.doi.org/10.3991/ijoe.v9iS1.2454

R.B. Neto[1], H. Fernandes[2], J. Pereira[1] and A.S. Duarte[2]
[1] Linkare T.I., Lda, Lisbon, Portugal
[2] Universidade Técnica de Lisboa, Lisbon, Portugal

***Abstract*—-e-lab is a remote laboratory infrastructure, powered by a specific software framework, that allows to perform experiments and retrieve data from a remote apparatus. The deployment of new experiments on this platform is a complex and demanding process and an integrated overview for it is missing. An overview of e-lab's specifications is presented taking into account software development requirements for the deployment of new experimental apparatus and possible improvements.**

***Index Terms*—e-learning, experimental education, on-line experiment, remote laboratory, university education.**

I. INTRODUCTION

Nowadays students show an increasing lack of motivation when learning Science. This suggests, among other things, the predominance of a classical approach and a missing contextualization of scientific concepts. With the overwhelming usage of new technologies by students it is necessary to promote e-learning as a normal part of the practice over time, complementing traditional teaching/learning methods.

Science is inherently experimental, and students can develop a much better empathy for the subject, and greater insight into its principles, if they are active participants in scientific discovery. However, for many different reasons, it is not always possible for students to perform experiments in the classroom or even by themselves.

On the other hand, when teaching physics or any other science, it is essential to create a strong bond of trust on all which is learned through online resources. Ordinary simulations (applets) often used to interpret contents more easily, belong to a "theory" area because they leverage on mathematical models and representations. Whereas practical activity ("observation") should be an important component for teaching science, specifically, repeating experiments and comparing between "observation" and theory and thus generating confidence in the results obtained.

e-lab [1] is a Remote Controled Laboratory (RCL) developed to bridge these two concepts: (i) incorporate the latest technology for distance learning, allowing to perform real experiments in a fully remote environment and (ii) allowing to watch them via video, as well as to collect all the experimental data in real time further increasing the correct confidence of its users on the performed experiment.

e-lab presents it self as a stable and very complete RCL platform, taking advantage of several years of development and investment through a partnership between the IST engineering faculty and Linkare T.I. R&D oriented SME., in Lisbon.

The deployment of new experiments in this platform is a complex and demanding process and an integrated overview is needed.

Also, improvements may be possible when analyzing this process between the technology being used and pedagogical objectives it serves.

An overview of e-lab's specifications and characteristics is presented namely to what concerns the full deployment of new experiments, taking into special account software development requirements and possible points of improvement.

II. E-LAB

In operation at IST since 2001, e-lab recently underwent a usability study, currently offering a simpler and more concise graphic user interface. Other developments were also made to further stabilized the software platform.

The deployment of new experiments largely depends on the recent developments that took place and so these will be specified and discussed.

A. Components

Currently e-lab's infrastructure is composed by several components:

(i) A central server to manage connections between the experimental apparatus and clients using ReC software platform (Remote experience Control). This platform was created with the purpose of managing a simultaneous multi-client connection to a multi-experiment remote laboratory. In order to do that it defines a generic standardized protocol of meta-information for data acquisition systems and a generic state machine for the execution of scientific experiments. Also is allows for quick integration of different services associated with Remote Controlled Laboratories (RCL), for this reason it is a central part for e-lab's laboratory.

(ii) A client application to be executed from the user side were major functionalities are accessible. The client application for experiments control has recently been redeveloped in order to offer a seamless user interface that may easily allow for the inclusion of new content related with an experiment. A brand new *Look-and-Feel* for this application was fully developed in Java and, as such, it is totally portable between different architectures or operating systems.

(iii) External applications like Quicktime or VLC for video, and other technical computing language for data

analysis (Origin, MSExcel, MatLab, SciLab, IDL, Octave and others). These allow for further exploration of acquisition results and the fact that e-lab's information structure already accounts for a solid implementation of exporting data in different generic formats applicable to these different kinds of tools: e-lab can already export tabular data from an experiment in CSV format and graphics in GIF, PNG or JPG formats.

(iv) Hardware Clients which accept connections for running experiments and are linked to each apparatus through a suitable driver.

(v) An apparatus driver unit for the appropriate translation between the high-level generic state machine commands and meta-information model used by ReC, to an experiment's specific communication and execution protocol – hardware controller tags or messages. The driver performs communication between the software platform (setup) and the hardware controller (microprocessor); it receives instructions from the platform and interprets and sends them to the hardware controller and it is also responsible for collecting the data from the hardware controller.

An apparatus driver needs to be defined every time a new experiment is to be deployed in e-lab and nowadays this can be more easily achieved by the development of a generic XML file that holds all the necessary specifications for an RS232 communication port, allowing for a quasi plug-and-play connection of the apparatus.

(vi) An hardware controller (microprocessor) prepared for data acquisition and control of each apparatus. It defines a specific communication and execution protocol for the whole experiment also allowing it to fit onto the generic software platform using the driver.

(vii) The apparatus itself equipped with appropriate sensors and motor devices to allow its full remote control and operation. With the implementation of the ReC generic RS232 driver, it became possible to use hardware from different sources and the process of adding new experimental apparatuses was simplified.

On e-lab most experiments are assembled using electronic junk re-utilized parts in order to approach an aspect that may considered very positive for schools which intend to deploy their own experiments.

(viii) A video streaming server to broadcast the video emission from each experiment over Real Time Streaming Protocol (RTSP).

The integration of video broadcast in the software platform and namely at the user interface level directly allows users to view the experiment's events coordinated with the configuration of the experiment and the reception of data.

At the moment e-lab has a graphic component available that allows the streaming of the video on MP4 encapsulation and coded in DivX, although other formats are supported. This is possible through a Java API (JVLC), although it has been discontinued recently and not all formats may be supported in the near future. Therefore this is an aspect that will require some development.

The native support for VLC is provided by Operating System (OS) dependent libraries and are directly distributed along with e-lab's interface application. Similarly e-lab delivers all the necessary DivX codecs which are installed, if necessary, along with the installation of the client side interface application.

(ix) A video encoder for each local experiment described by its session description protocol file. This characteristic allows for great improvement regarding the usage of video as a tool for the experiment itself, for instance allowing the use of different video formats or future implementations of video capture coordinated with the experiment itself.

(x) An external site were experiments are described and the knowledge database exists. A full wiki-database containing the description of the experiments and support educational content has recently been developed allowing for further improvement of production and sharing of content between users and developers. It also contains important content regarding the development and deployment of new experiments.

Additionally, in the future this knowledge database can be further integrated onto the user interface itself improving its browsing and consultation properties during the execution of the experiment, allowing a closer structure to a full Learning Management System (LMS) environment

(xi) A video-conference infrastructure based on Openmeetings in support of Webinars and for b-learning monitoring. Although e-lab's structure and interface has implemented full ability for sharing of data and chat between users on a same experiment or laboratories, further video conferencing ability may be useful. For this Openmeetings is made available and the usage of e-lab through Openmeetings is straightforward.

(xii) A scheduler (reservation system) for experiment authorization to main users. In order to allow for greater integration with Learning Management Systems (such has Moodle) and aiming at e-learning and b-learning full capabilities, e-lab has implemented an authorization and authentication service that allows to quick use of existing user databases in order to create a login based laboratory with usual scheduling options, allowing for instance for teachers to reserve in advance an experiment for their class, during which no other user will be able to control that experiment.

(xiii) Several external Moodle authentication and content management servers. In the last couple of years an association between IST and the Portuguese Ministry of Education has implemented a full e-lab Integrated Multi-Moodle authentication with several Moodle servers for access and content management aiming at supporting science education for basic third cycle and secondary education levels.

On e-lab, the user and password is the same as in the Moodle the user has access to. e-lab achieves this by integrating with the reservation system and delegating it's authentication and authorization requests, making it simple to integrate other authentication sources as required. e-lab doesn't currently provide integration with SAML products or OpenID or OAuth providers, but these options may be implemented in the future.

B. ReC

The main application behind e-lab is a software framework in a client-server approach. As mentioned, by defining a generic standardized protocol of meta-information for data acquisition systems and a generic state machine for the execution of scientific experiments, this platform manages a simultaneous multi-client connection to a multi-experiment remote laboratory. It also allows integration of different the services mentioned (sharing data, chat, authentication, repository,…).

This server empowers developers of Data Acquisition and Robotics Control Systems to put their local software to work over any network: high speed intranet LAN, WAN or simply the Internet. It is based on standards such as Java & CORBA & XML [3], so it may also be integrated with any legacy code quite easily. The main interfacing system is open, and so it may be extended to suite the needs. It works on all the operating system platforms: such as Unix (and Linux), Solaris, Microsoft Windows, and Macintosh although the present e-lab is running over Gentoo Linux [4].

The implementation os ReC is composed by three main distinguishable parts:

(i) Harware Server – Responsible for the exposition on the platform of the native data acquisition system (hardware) operations; also responsible for the communication between the hardware drivers and collection of data and its transmission to the MultiCastController.

(ii) MultiCastController – Responsible for distinct instantiations and expositions of Hardware Servers and their access in a controlled, secure and coordinated manner with the client side. This component also integrates repository abilities and persistent maintenance of data for all the apparatuses controlled by it. This part is central and critical for its requirements of scalability, volume and stability of the entire platform.

(iii) HardwareClient – Responsible for communication with the MultiCastController, integration of different graphic interfaces for configuration and data presentation in a common and coherent interface, independent of the OS being used.

It is over this ReC structure that may be developed different drivers, for different experimental apparatuses, has well has different graphic interfaces when deploying a new experiment. *Figure 1* shows a schematic representation of e-lab's structure.

C. *Laboratory Infrastructure*

The current laboratory exists in an open-space accessible to public view at IST trough a glass wall and there are more than 20 apparatus, not all of them available online.

Each PC is member of a cluster system and typically drives two experiments. This approach becomes similar to other Mutil-Server approaches, which require a server for every experiment, but requiring less physical space, energy consumption and financial effort.

The main CPU usage is related with video encoding, taking one core at 90% for this purpose.

The cluster system is composed by blade PCs nowadays being migrate to Intel Atom with IG RAM and 12V multi-rail power supply. In this topology only the servers, both ReC and video streaming need a high-end processor and are not Atom-based machines. Presently four main server machines exist: (i) the experiment server controller and the application server; (ii) the video server; (iii) the cluster main server for the archive file system and boot execution managing and (iv) the videoconference web tool based on a browser client flash application.

An internal private intranet is used for interconnecting the cluster blades and only the servers are exposed to the outside internet connection.

To this topology it's also being added a dedicated server to allow support for future experiment development: a Production Server.

As mentioned, regarding software development, when creating a new experiment one needs to develop a driver for the new hardware and a related User Interface, these will have to be tested repeatedly, during development, before being made available to public, this is only possible after deploying the full experiment in the platform. Up until now there existed only one server system where this is was possible, the same system where all the functional experiments are available. As one would expect, One of the main difficulties encountered when developing a new experiments under development were common errors that would lead to sporadic serious crashes or interferences of all the connected and correctly running experiments. The Production Server now being implemented will allow for a permanent development environment where new experiments may be deployed for testing before being permanently deployed on e-lab's servers open to the public.

This change in method will also allow for a great improvement of conditions for developers and extreme reduction of testing time and requirements.

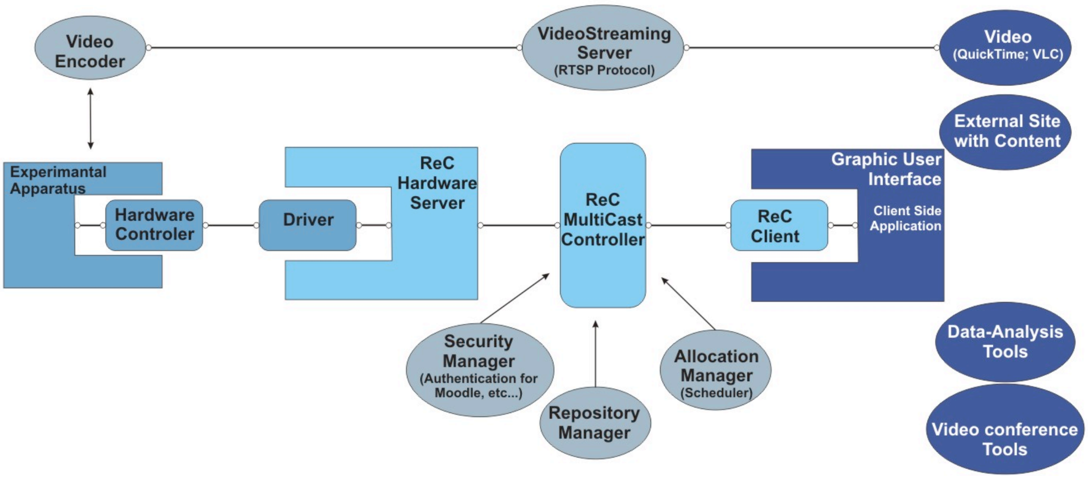

Figure 1. Schematic overview of e-labs components and their elations.
To the left side is represented the Hardware end of e-lab's platform, whereas to the right side the User end is represented.
In the middle, managing connections, information transfer and services between the two lies the ReC generic platform.

D. Apparatus controller

In effect RCLs needs very specialized and flexible hardware which may be very difficult to accommodate with off-the-shelf hardware. A dsPIC controller has been elected and C programming adopted. Currently most of e-lab apparatus can be operated with this in-house developed Eurocard board assisted with customized mezzanine boards, typically connected in a piggy-back arrangement. This mezzanine board houses signal conditioning circuits, power drivers and other major specialized circuits.

III. EXPERIMENTS DEPLOYMENT

This overview work is based on two new experiments that are in the process of being deployed in e-lab's laboratory: the "Pendulum" and the "Pin Hole" experiments.

RCLs like e-lab could be a precious help to provide important tools for higher-level students The experiments that are deployed are aimed at post secondary level and first year of University in a perspective of curricular integration. Also, content and experimental protocols are developed aiming at integration and use for basic engineering classes at IST.

Moreover RCLs can be understood as data generators for a broader learning environment. e-lab should, therefore, be offered as an integrated tool merged with other products for better use, for instance equation fitting spreadsheets and data analysis tool packs.

In order to offer RCLs in such a way that any student at a university or pre-university levels can easily use it, a greater effort has to be made in simplifying technology usage and computer literacy. Also to this concern one of the perspectives that e-lab offers is the ability to allow students from these levels of education to develop and deploy their own experiment at their schools, and it is to this matter that greater developments still need to take place.

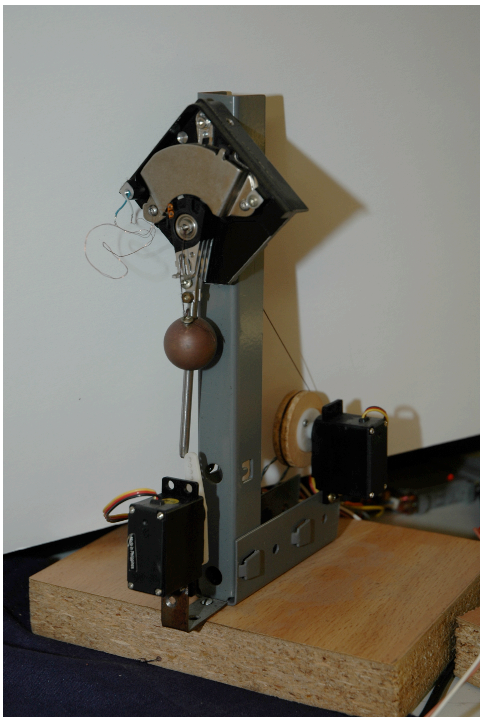

Figure 2. Experimental apparatus of the Pendulum Experiment to be deployed on e-lab.

As one can conclude from the presented overview the deployment of experiments in e-lab is oriented for a plug-and-play approach, both regarding the plugging of new hardware as well as the development of new interfaces but such a full plug-and-play solution is still far from being implemented.

As mentioned in order to connect a new hardware to e-lab's current framework one needs to develop a suitable driver for it. Also, if the experimental apparatus to be connected needs it, a new Graphic User Interface has to be developed.

The process of deploying a new experiment occurs in several steps, which are briefly described. The process of development of hardware and programming of the hardware controller (microprocessor) is not included in this description, only the necessary steps to implement the two aforementioned pieces.

A. Creating the structure for a new experiment:

On e-lab's servers a very well defined structure of files allows for the correct implementation and simplification of the development process. This structure needs to be replicated for every experiment in order to allow correct compilation of the code. As such a command line allows for automatic creation of a new directory with the required file structure and small "read me" files exist in everyone of the sub-directories in order to ease comprehension of it's content.

This structure will contain not only source files to run the experiment correctly, but also all the information related to the experiment, such as documentation on the hardware – like the microprocessor's functioning – protocols, apparatus description, datasheets, etc… related to the experiment.

B. Configuring the server/driver:

This is done by the configuration of two XML files:

(i) HardwareInfo.xml – contains the specification of the apparatus characteristics to be used by ReC.

Specifies an hardware configuration, e.g. if the ReC's Hardware Description Model describes that a specific variable can have range from 0-5V or from 0-12V the Configuration Data Model in this file will specify which of these ranges is actually being used for a given experiment. Or if, in general, an interface can measure with maximum 8 sensors, but for a specific experiment it is limited to 2.

(ii) Rs232Config.xml - contains the specification of the textual protocol to communicate with the apparatus. As mentioned for an RS232 port all which is needed to define communication driver is to create one of these new XML files with the correct specifications for an experiment.

Both these files must respect very strict ruling in order to be later used by ReC's Hardware Server, MultiCast-Controller and HardwareClient. For instance respecting its generic state machine implementation. As such some knowledge of ReC as well as XML is required for this and, although it may not be considered too demanding, this is perhaps a matter which can still be target of future development. For instance with the development of a an interface which would allow to personalize the XML descriptor directly for each experimental apparatus. The development of such a descriptor interface would turn the development and deployment process on e-lab extremely simpler and straight forward as well as much more secure.

One other point of improvement could be the development of such generic XML driver files for other types of ports such as USB.

C. *Configuring the GUI:*

The GUI provides the user with the functionalities related to the customization of the experiment's parameters, as well as the presentation of the results of the experiment's execution.

The most recent developments to e-lab's GUI created an entirely new Java based *Look-and-Feel* based on a framework approach for the creation of GUI for different experiments. This means e-lab's user interface is built based on the premiss of small basic components that can be (re)assembled together, although not in a very automatic and simple enough way, in order to do it one still requires some knowledge: Solid Java programming language knowledge, Swing API theory and practice and Model View Controller pattern. But using specifications such as JavaBeans enables the construction of new user interfaces by drag-and-drop, personalization and serialization in a much more simple way.

But again there is also a need for a descriptor to enable integration of the user interface components in every experiment. e-lab also prescribes a format for this plugin descriptor, enabling the construction of authoring interfaces which in most cases are not more than just interfaces to personalize the XML descriptor for each experimental apparatus.

1) *Creating the Customizer:*

A Customizer is a graphical component that handles the user input for each configuration parameter in order to define an experiment environment. Each experiment has one single Customizer to enable the definition of valid configuration values according to the experiment specification.

This is a piece that needs to be created anew, for each experiment, with their own specific characteristics and customization parameters

2) *Creating Displays:*

The displays are the components for presenting the experiment's results. Those results may be presented as Graphics; Tables ; Videos; or any specific gauge, sensor or other display forms (could even play a sound, for instance). To simplify the displays development some default implementations of Data Tables; Charts and Sensor Gauges are provided on the framework.

3) *Configuration files: ReCFaceConfig.xml*

This file contains general information related to visual configuration of the experiment, allowing one to configure, for instance, if there is a video associated to the experiment, used to show the execution of the experiment in real time.

4) *Internationalization*

To adapt the experiments user interface to various languages all the UI resources must be placed in a resource bundle file. Usually, there are two messages files: *messages.properties*, with the portuguese translation of the experiment and *messages_en.properties*, with the english translation of the experiment.

D. *Deploying*

Finally the experiment can be built or packaged and then it's deployment can effectively take place, in which it is necessary to enable the experiment in e-lab's GUI and perform driver startup, making sure the hardware is properly connected to respective port.

This process is usually carried out by Linkare T.I. given the previous risk mentioned, of deploying an experiment with untested bugs within it's code, possibly dangerous ones. The creation of a Production Server will help improve this deployment method and further simplification can be achieved here in the future by taking into account possible simplifications on the previous steps.

IV. .CONCLUSIONS

e-lab is an RCL built on an approach aimed at scalability and simplified usage. Although these aspects are already quite advanced nowadays allowing users to perform experiments on-line with very good usability and allowing them to take advantage of different tools at their disposal to further improve their experience, the fact remains that the underlying RCL framework can still be object of very significant and important developments to what concerns usability and scalability.

The greater simplification of the deployment of new experiments can rapidly lead to the significant increase of e-lab's experimental apparatus offer and to it's attractiveness, allowing it to go beyond it's limited window room at IST and become connected to experimental apparatuses developed elsewhere by students and teachers, along with new related content. The development of an XML based descriptor for an experiments driver, the undergoing implementation of a Production Server and the recent implementation of an associated wiki are very important steps to achieve this simplification already and further work on these areas, aiming at even more simple and graphical ways to develop and deploy a new experiment should be pursued.

ACKNOWLEDGMENT

Special thanks to Prof. Horácio Fernandes, Eng. José Pedro Pereira, Eng. Paulo Ribeiro.

REFERENCES

[1] e-lab's main website, <http://elab.ist.utl.pt>.

[2] e-escola main website <http://www.e-escola.pt/elabexp.asp>.

[3] Java website, <http://www.java.com>.

[4] Gentoo's Website: <http://www.gentoo.org>.

[5] ReC – Remote Experienced Control, <http://www.linkare.com/product/ e-lab>, 2012.

[6] dsPIC30F Family Reference Manual, available on-line at <http://www.microchip.com>, 2011.

[7] Remotely Controlled laboratories – RCLs, <http://rcl.physik.uni-kl.de/>, September, 2011.

[8] R. B. Henriques, A. S. Duarte, H. Fernandes, T. Pereira, J. Fortunato and J. Pereira, "Generic protocol for hardware control @ e-lab"

[9] Linkare's manuals on ReC.

AUTHORS

R. B. Neto is with the Linkare TI - Tecnologias de Informação Lda, Av. Duque D'Ávila 23, Sala 06.A, 1000-138 Lisboa (e-mail: rneto@linkare.com).

H. Fernandes, is with the Associação Euratom/IST – Instituto de Plasmas e Fusão Nuclear, Instituto Superior Técnico, Universidade Técnica de Lisboa - 1049-001 Lisboa – Portugal (e-mail: hf@ipfn.ist.utl.pt).

J. P. Pereira is with the the Linkare TI - Tecnologias de Informação Lda, Av. Duque D'Ávila 23, Sala 06.A, 1000-138 Lisboa (e-mail: jpereira@linkare.com).

A. Duarte is with the Associação Euratom/IST – Instituto de Plasmas e Fusão Nuclear, Instituto Superior Técnico, Universidade Técnica de Lisboa - 1049-001 Lisboa – Portugal (e-mail: aduarte@ipfn.ist.utl.pt).

This work was supported by Instituto de Plasmas e Fusão Nuclear, Instituto Superior Técnico, Universidade Técnica de Lisboa and Linkare,T.I. SME. It is an extended and modified version of a paper presented at the International Conference on Remote Engineering & Virtual Instrumentation (REV2012), held at University of Deusto, Bilbao, Spain, July 4-6, 2012. Received 17 November 2012. Published as resubmitted by the authors 18 December 2012.