

Smart Adaptive Remote Laboratories for Education (Implementation of a prototype)

Luis Felipe Zapata Rivera, M.S.
Computer & Electrical Engineering &
Computer Sci. Dept.
Florida Atlantic University
Boca Raton FL, USA
L.F.ZapataRivera@ieee.org

Leonardo Ribeiro Da Silva
College of Computer Engineering
Centro Universitário Salesiano de São
Paulo
Campinas, São Paulo, Brazil
leonardo.silva@sj.unisal.br

Maria M. Larrondo Petrie, Ph.D.
(Mentor)
College of Engineering and Computer
Science.
Florida Atlantic University
Boca Raton FL, USA
petrie@fau.edu

Abstract— With the development of Internet of Things (IOT) devices, applications, advances in internet communications, web and mobile applications, and specially with the increasing demand for online education around the world. Emerge a new challenge for the practical component of the learning process. The use of online laboratories is an alternative for those schools that cannot afford expensive laboratory equipment or for those institutions that offer online education and their students are geographically spread around the world. Online Laboratories are classified as virtual labs and remote labs [1]. The SARL architecture [2] defines the use of remote laboratories integrated with VLE's providing the users an individualized experience with the labs, using the user information to adapt his experience in terms of complexity and topics, and allowing the teacher to parametrize the labs activities.

This paper shows the implementation of a prototype of this type of systems. The prototype implements some of the 3 components of SARL system, including the use of one experiment controlled by a raspberry pi, a web camera and a PHP application that control the user access and the experiment selection (RLMS), and the integration of the system with the VLE (LMS Moodle) is also included in the prototype.

Keywords—Remote Laboratories; Adaptive Interfaces; VLE; LMS; Remote Laboratories Management System

I. INTRODUCTION

Online Laboratories are classified as virtual labs and remote labs [1]. Virtual labs are related with simulation of physical experiments, normally using 2d or 3d software environments. These labs are helpful when the cost of running each experiment in the real equipment is too high or when performing the real experiment is too dangerous, generating risk for people or for the physical location. An example of this, are experiments including chemicals that can explode. The virtual labs must behave like the real experiment taking in account as much it can the variables involved in the process. Some of the virtual labs are known as batch experiments, those use real data taking from actual measurements that serves as inputs of the simulation. The other type of online laboratories are the Remote laboratories, these labs use real physical labs and through the use of internet allow the user to control the lab equipment letting the user perform the experiment remotely.

II. ARCHITECTURE OF RLMS

The Remote Lab Management System is the platform integrated with a Virtual Learning Environment (VLE) that manage the remote lab resources, user access and remote lab interfaces, accordingly to the user that is connecting. This platform is implemented in a server separated of the laboratory experiments, to have a better performance to the interaction of the user.

To have a smarter and efficient adaptive controller, the architecture of the RLMS consist of an access management, resource management and a smart adaptive controller.

The **access management** of the RMLS application consists of a web interface for login that will receive the user credentials and check his permissions to give him access to the platform. According to the role, the user will have different rights inside the system. For instance once the session is open for a student, he will have access to some experiments, but as a teacher connecting, he will have access to the resource management functions.

Once the user is authenticated with the right permissions, he can check the **resources** that are available (not in use by another users or in maintenance) in the platform and then get started with the activities that were proposed in the platform. Accordingly to the permissions of the user, he can do the activities of the lab, for example write a code at the platform or play with the buttons, this commands will be sent via PHP programming language through the internet to the **controller** that will execute these commands and send the information to the physical experiment. According to the user activities the **smart adaptive** section will record every step of the user interactions and then adjust the view for different activities and assessment.

The experiment **controller** is a credit card-size computer board with Debian operational system that is capable to communicate with the physical experiment, sensors, camera and any other commands via Python programming language.

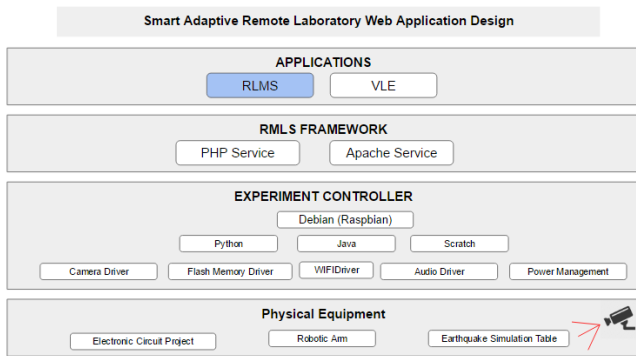


Fig. 1. Design Architecture of RLMS.

III. EXPERIMENT PREPARATION

For this prototype, a simple experiment controlled by a Raspberry Pi3 via GPIO pins was implemented based on LED's, making use of a breadboard, resistors and cables. The idea is to have control over the lights using three output pins of the Pi3 to perform different tasks.

The breadboard was one of the most important pieces of the experiment to build a circuit without requiring absolutely no soldering. The cables were used to make the connections of the pins, LED'S, and resistors in the circuit.

To be part of a remote laboratory setup, laboratory experiments need to be adapted to support the remote access and control, for that purpose an architecture based on the use of the Raspberry Pi 3 (RPi3) hardware was used. The WebIOPi [3] service has been installed in the RPi3 to expose the General Purpose Input Output (GPIO) communication ports to the web. The service package called Motion [4] was used for video streaming server. With the controller running all the services for communication and the video streaming, the process will be completed with the connection of the experiment to the controller.

The adaptive component in the implementation choose between two possible activities with the laboratory. The first one is a simple led blinking task and the second requires programming the LED's to do a sequence red, yellow and green during ten times. The second activity asks the student to create a dimmer light sequence, for that the student needs to know the concept of Pulse Width Modulation (PWM), the student must write the program using Python language. The code is saved, validated and executed in the remote experiment in real time. According to the student characteristics, one or the other activity is required.

IV. RLMS DESIGN AND IMPLEMENTATION PROCESS

The design of RMLS implies the implementation of a Virtual Learning Environment – to manage the user access to the content and the assessment; a web framework to manage the learning content and interact with the users; a backend server to communicate with the operation system of the computer board

Pi 3; a backend server to communicate with the output pins to control the experiments – responsible to send and receive commands to the experiment – and a streaming service to monitoring the activities from the experiment via a web camera. To have a better performance, the environment was implemented in two different services: web services host – that was installed all the VLE and framework environment – and the Raspberry Pi 3 host – that was installed the operation system to communicate with the experiment.

Web service environment

To configure the environment to implement the LMS and the RLMS framework was used an Amazon web service. The Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides secure and scalable computing capacity in the cloud. The implementation of this web service reduced the time required to obtain a new physically server instances installation and provides the main tools to create different scenarios in the cloud.

The Amazon EC2 provides the following features used by the RMLS environment: Virtual computing environments, known as *instances*; Various configurations of CPU, memory, storage, and networking capacity for the instance; Secure login information for the instance using key pairs (AWS stores the public key, and the RMLS instance store the private key in a secure place); Multiple physical locations for the instance data using Amazon Elastic Block Store (Amazon EBS); a firewall that enables the instance to specify the protocols, ports, and source IP ranges that can reach the RMLS instance using security groups and static IPv4 addresses for dynamic cloud computing, known as Elastic IP addresses [5].

Within this scenario was installed the operational system – Ubuntu Server 16.04 LTS (HVM), SSD Volume Type – that is compatible with all the resources that was needed to implement the platform. With this environment installed and configured the environment is prepared to run safely any web application.

Communication resources

With the webserver prepared and configured a few more resources are required before start the VLE and RLMS implementation. These resources will be responsible to record all interaction data, establish a communication between the database and user interaction and a backend programming language to communicate with the servers. For this prototype, the resources installed to manage and control data information are Apache server, MySQL and PHP.

Apache implementation

The collaborative software development **Apache** is a HTTP (Web) server that allow the communication and manage all data request between the client (web browser) and web server host. Using the protocol HTTP the Apache server is reliable, secure and most common used for web applications [6]. The version installed on this project it was Apache http 2.4.25 – updated 2016-12-20.

PHP implementation

The **PHP** is responsible to start and end all processing instructions from the web browser that will be executed on the

server, generating a HTML which is then sent to the client (web browser). The advantages using PHP are that it is extremely simple for a beginner, and in the same time offer many advanced features for a professional programmer. The main reasons to choose the PHP for this prototype it was the fact that PHP allow to collect form data, generate dynamic page content, send and receive cookies from the client. Besides it, with PHP the project is not limited to output HTML. PHP's abilities includes outputting images, PDF and XML files. These files can autogenerate and save them in the file system, printing it out and forming a server-side cache for the dynamic content [7].

PHP can be installed on all major operating systems, including Linux, Micosoft Windows, Mac OS X, Risc OS, many Unix variants and others [8]. The PHP requires web server environment pre-installation to be used – for this prototype was used Apache – and the PHP version installed for this prototype was PHP version 7.

Database implementation

Database is a structured collection of data that will allow to record data and interactions of the user executed on the RMLS system. To have a good performance of the system, it was implemented a database relational to stores the data in separate tables rather than putting all the data in only a single table. The logical model, with objects as databases, tables, rows and columns, offers a flexible programming environment.

In this database will be recorded the user member's information's, tasks, activities, grades and any data generated in the system. All these data are used to control the user access, tasks corrections, students' progress, reports, output view and general management of the system. The database system installed for this prototype it was MySQL.

Virtual Learning Environment

For this prototype scenario, the Learning Management System platform chosen was the Moodle LMS. This open source platform for educators is a modular system based on plugins that can be customized to build whatever is needed for this platform [9].

With this flexible open source LMS – Moodle version 3.0 – the platform allows the configuration of the LTI connection between the RLMS and the LMS. The teacher can easily integrate the labs in the course and define the parameters for the Smart Adapter directly from the LMS. Figure 4 shows an experiment composed of a set of LEDs that controls a servomotor, simulating a traffic signal. This experiment is integrated in the LMS Moodle.

The next two figures 2 and 3, show adapted views for the same experiment. In the first screenshot shows the authentication interface in which two students (A and B) see different version of the same remote experiment according with their own information, Student A is asked to do two activities.

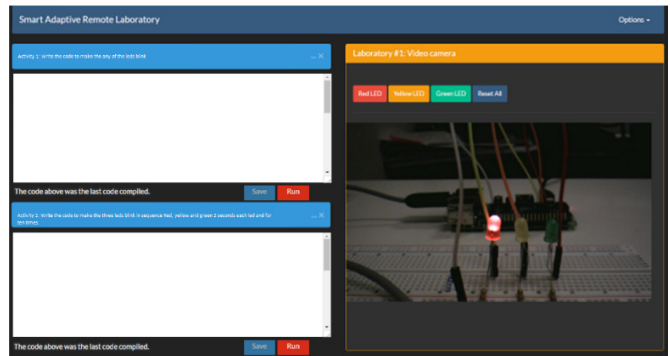


Fig. 2. SARL- Remote lab for a LED's experiment adapted to student A.

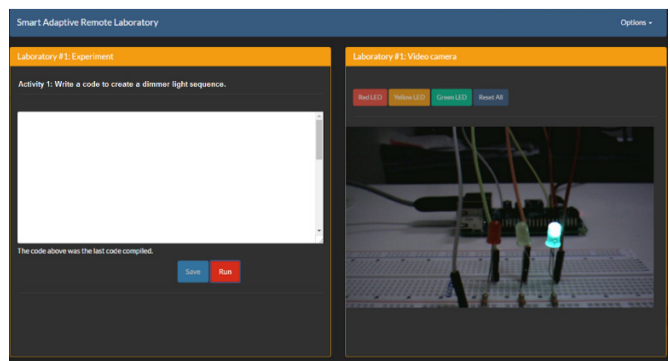


Fig. 3. SARL- Remote lab for a LED's experiment adapted to student B.

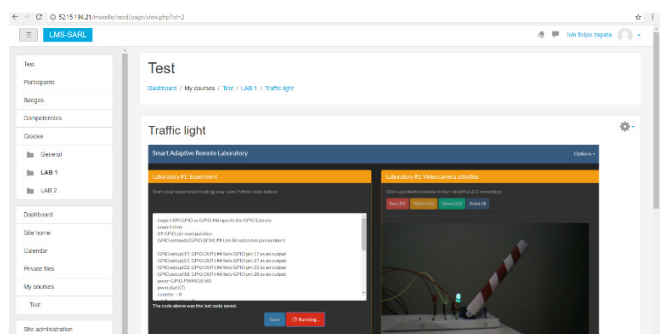


Fig. 4. SARL Registration and login interface.

Streaming implementation

The RMLS system prototype allow the user to control a experiment remotely. This interaction and the activities of the experiments is showed in a web video interface that can streaming images in real time. For this prototype was installed a streaming service called motion [4].

Motion is a program installed at RMLS system that monitors the video signal from remotely camera that are monitoring the experiment. The motion can detect if a significant part of the picture has changed – for this implementation was configured to capture 30 frames per seconds – It means that with this program the experiment can be viewed by the user via a web camera.

V. CONCLUSIONS

This prototype show the power of the remote labs laboratories technology when is used with the most recent web tools, modern programming languages and integrated with learning environments systems. In the near future, this project will show a complete implementation for the real use of the tool in academic environments, presenting the findings after the use of the system by a group of students.

REFERENCES

- [1] D. G. Zutin, M. E. Auer, C. Maier, and M. Niederstatter, "Lab2go – A repository to locate educational online laboratories," in Proc. of IEEE EDUCON 2010 Conference: The Future of Global Learning Engineering Education, Madrid, Spain, 14-16 April 2010, IEEE, pp. 1741-1746. doi: 978-1-4244-6571-1/10.
- [2] L. F. ZAPATA RIVERA, M. M. LARRONDO PETRIE, "MODELS AND SMART ADAPTIVE INTERFACES FOR THE IMPROVEMENT OF THE REMOTE LABORATORIES USER EXPERIENCE IN EDUCATION," PROCEEDINGS OF THE 13TH INTERNATIONAL CONFERENCE ON REMOTE ENGINEERING AND VIRTUAL INSTRUMENTATION (REV 2017), NEW YORK, NY, USA, IEEE, 15-17 MARCH 2017.
- [3] WebIOPi. GPIO API for Raspberry Pi [Online]. Available: <http://webiopi.trouch.com/>
- [4] Motion – Motion detection Web Camera service [Online]. Available: <http://www.lavrsen.dk/foswiki/bin/view/Motion/WebHome>
- [5] Amazon Elastic Compute Cloud Documentation [Online]. Available: <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html>
- [6] Apache –HTTP Server Project [Online]. Available: <https://httpd.apache.org/>
- [7] Welling, Luke, and Laura Thomson. PHP and MySQL Web development. Sams Publishing, 2003.
- [8] PHP Documentation - General Installation Considerations [Online]. Available: <https://secure.php.net/manual/en/install.php>
- [9] Rice, William, Moodle E-Learning Course Development - Third Edition. Paperback – June 25, 2015.