Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnline
CONFERENCE PAPER ARCHIVE

IFAC-PapersOnLine 49-30 (2016) 086–091

Design and implementation of a remote lab for teaching programming and robotics

João Paulo Cardoso de Lima * **Lucas Mellos Carlos** *
José Pedro Schardosim Simão * **Josiel Pereira** *
Paulo Manoel Mafra * **Juarez Bento da Silva** *

* *Federal University of Santa Catarina, Araranguá, SC, Brazil (e-mail:joao.pcl,lucas.mellos@grad.ufsc.br; pedro.simao,j.p.josiel@posgrad.ufsc.br; paulo@mafra.eti.br; juarez.silva@ufsc.br).*

Abstract: In this document we describe the development of a remote lab and two user interfaces for teaching programming and fundamentals of robotics using Arduino boards. One of the interfaces uses visual language to make the experience more attractive for those who are beginning their studies. The second interface provides an environment similar to the original Arduino's one for those users who have some background to use textual programming language. In addition, we present an usability evaluation of the application using visual programming language to obtain a broad feedback mainly in terms of target audience's acceptance and suggestions for improvements. The environment is still an ongoing work, but it is suitable for courses and workshops, especially for institutions with limited resources.

Keywords: Remote laboratories, computer programming, educational robotics, visual programming language, Arduino.

1. INTRODUCTION

There is a growing interest in teaching computer science concepts for young people. This interest is not only due to the training for a future job or to the high demand in this field, but also due to the benefits that computer programming can bring to secondary education curriculum. Computer programming and robotics, for instance, play an important role in the development of skills such as problem solving, computational thinking and creativity (Sáez-López et al., 2016).

The majority of pedagogical approaches for developing technological competence and programming skills in schools are based on theoretical, illustrative classes. On the other hand, robotics enable approaches closer to actual applications and concepts from engineering and mathematics fields (Tocháček et al., 2016). However, robotics require classes equipped with appropriate laboratories and experiments that follow the paradigm "learning by doing" to ensure deep understanding (Jara et al., 2011). Nonetheless, the majority of the existing solutions for teaching programming and robotics is not so affordable, mainly when we consider developing regions or educational institutions that suffer from shortages or inadequacy of educational materials, physical infrastructures and other resources.

The lack of updated, appropriate labs are the main hurdle in teaching subjects related to science and engineering. Practical activities in schools are as important as they are for higher education. However, the index of lack of laboratory infrastructure and practical activity are even higher for basic education. For instance, when it comes to Brazilian public education system, only 8% of schools have their own science lab and 45% of them have computer lab according to the 2014 census results gathered by INEP (2014).

This documents aims at sharing the results that the authors have obtained during the development of a remote lab and two client applications for teaching programming and robotics with Arduino. The main difference between the two applications is on the graphical user interface (GUI); one of them uses Visual Programming Language (VPL) on a mobile application and the other uses Textual Programming Language (TPL) on a Web application, which is similar to the Arduino Integrated Development Environment (IDE). The authors also asked the target audience to evaluate usability criteria and utility of the VPL application, since it slightly differs from proposal of the Web version.

The paper is organized as follows: the role of computational thinking and laboratories are discussed in the first section. Then, we present the prototype, which includes two GUI, a server application and a physical lab in the second section. Afterwards, the usability evaluation carried out with students and teachers is described in the third section, the results are discussed as well. Finally, final considerations and future works are discussed in the fifth section.

2. THE ROLE OF COMPUTATIONAL THINKING AND LABORATORIES IN ENGINEERING EDUCATION

Since 1960, with the advent of programming languages with educational purpose such as Logo and Basic, re-

10.1016/j.ifacol.2016.11.133

searchers have begun to explore how computer programming could be integrated into courses in mathematics, physics, chemistry, biology and engineering. Furthermore, it was not only intended for hard sciences and engineering, but also for history, art and music courses because the main idea was that programming languages might have a positive impact in all education. These efforts are still alive today through projects such as Scratch, Lego Mindstorms and Arduino (Lye and Koh, 2014).

When students are programming, they are being exposed to computational thinking. The computational thinking is the thought process involved in the formulation of problems and expression of their solutions as information that a computational agent can effectively perform (Wing, 2011). This term popularized by Wing (2006) will be considered a fundamental skill for people around the world in the mid 21st century as much as reading, writing and knowing how to perform arithmetic operations are essential skills today.

Computational thinking is an fundamental competence for most engineering professionals. Engineers will be responsible for the design, creation and analysis of even larger and complex systems, requiring effective communication with interdisciplinary teams. In addition to supporting the acquisition of knowledge and development of problem solving skills, computational thinking can also influence social skills, such as improvement in communication and also improvement in translation of real-world problems to the technical domain and technical issues back to common language (Gross et al., 2014).

On the other hand, since the beginning of engineering education, laboratories have taken a central role in engineering training. It is important to distinguish three basic types of engineering labs: development, research and educational one's. When students, especially undergraduate, go to the laboratory is not to extract a data particular need for your project or find some additional knowledge to the humanity, but to check something that practicing engineers already know, that is, to be able to use instrumentation, analyze mathematical models, analyze data, learn from their mistakes, and others (Feisel and Rosa, 2005).

According to Gomes and Zubía (2008), remote labs are considered a good alternative to meet the demand for laboratory activities in engineering courses, in addition to providing access remotely without restriction of location and time. As stated in Alves et al. (2007), a remote lab corresponds to a situation where the control and observation of physical instruments and objects are provided via a computer network. Providing remote access to laboratories is a straightforward idea for distance learning. However, remote access also offers a simple solution to problems such as distance, expensive equipment, idleness and availability of resources (Cooper and Ferreira, 2009). In the literature it is generally found remote laboratories based on programmable logic controllers (PLCs), programmable logic devices (PLDs), field-programmable gate array (FPGA) and microcontrollers for teaching robotics, control, digital circuits, and so on. (Tawfik et al., 2012, 2013).

On the other hand, classes that teach programming and robotics using Lego, Arduino, Scratch, among others, are becoming increasingly common and indicate positive results even in teaching concepts related to the STEM fields (Benitti, 2012). Furthermore, the combination of robot programming and remote laboratory attempts to abstract complexity and difficulties in the setup, and focuses on the development of logical reasoning to solve real problems using resources shared among the users via the Internet.

Although there are some Arduino programming environments with similar purpose, the environment presented here differs from all others because it provides GUIs focused on usability and user mobility. Though, users may use this solution to program and test their code in real boards, sensors and actuators, experiencing all the features available in the Arduino IDE. In addition, it not only provides activities suitable for both basic and higher education, but it intends to progressively increase difficulty in order to engage students since the first use.

3. PROTOTYPE

The prototype was developed in Remote Experimentation Laboratory (RExLab), which also provides other remote labs related to STEM areas for secondary and higher education. The Arduino lab is part of a set of cost-effective labs available on the learning environment called RELLE [1] , which stands for REmote Labs Learning Environment, also developed by RExLab. .

The Arduino lab can be reached either through a mobile application or Web browsers. The GUI based on VPL is available for Android devices and web application for browsers as well, but the TPL interface is only available as web application. However, both interfaces adjust for different screen sizes, which increases application compatibility to multiple devices and thus makes it possible to achieve different users who are included in the most diverse realities.

The two applications follow the same sequence of steps. Users may program and interact with an Arduino Board and a set of electronic components either using block language or Arduino language, which is based on C/C++. In a short, the user creates a code and compiles it, and the output message of the compiler is shown. Then, the user may send the program to run on the microcontroller remotely, the output message of the uploader is shown, and serial terminal may be used to exchange data between the GUI and the board. In addition, a video streaming is shown while the board are running his/her program. The entire sequence is shown in the Fig. 1.

3.1 Client applications

The two interfaces to access the experiment were created using web technologies and responsive design to enable users to have the same experience on devices with different screen sizes. Both environments have code editor, message area, menu, buttons for common functions, video streaming, text console and a circuit diagram set previously on the institution that hosts the remote lab. Moreover, some examples are provided using different electronic components available, and functions for handling files are provided as well.

[1] http://relle.ufsc.br