Zeszyty Naukowe Wydziału Elektrotechniki i Automatyki Politechniki Gdańskiej Nr 37

I Konferencja

e-Technologies in Engineering Education eTEE'2014

Politechnika Gdańska, 30 kwietnia 2014

ONLINE ROBOTIC LABS IN SOFTWARE ENGINEERING COURSES

Fernando GONZALEZ [1], Janusz ZALEWSKI [2]

1. Dept. of Software Engineering, Florida Gulf Coast University, Ft. Myers, FL 33965, USA
   tel.: +1 (239)590-7823 e-mail: fgonzalez@fgcu.edu
2. Dept. of Software Engineering, Florida Gulf Coast University, Ft. Myers, FL 33965, USA
   tel.: +1 (239) 590-7317 e-mail: zalewski@fgcu.edu

Abstract: Florida Gulf Coast University, College of Engineering has developed over recent years a sophisticated undergraduate software engineering lab for use in embedded systems and related team project courses. A number of teaching modules have been developed, with emphasis on security, complex systems, and web-based access. The objective of the current project is to focus on developing a lab specific to robotics applications, with online access, which can be used across the undergraduate curricula, especially in software engineering courses. Creating an Online Robotics Lab takes advantage of the existing equipment, software and curricular modules to expand them for broader use, and help develop a more advanced full-scale operation across the curriculum. Developing such a lab will have an impact on advancing teaching in disciplines such as Sciences, Technology, Engineering and Math (STEM).

Keywords: Online labs, robotics labs, STEM curriculum, software engineering education.

1. INTRODUCTION AND BACKGROUND

Robotics has become a major application and research area in technology in the 21st century. Robotic technologies are being used in automation in almost every aspect of our lives, from automatic doors to assembly lines to space missions. Correspondingly, robotics education is also advancing, both in the U.S. [1] and worldwide [2].

Following this trend, robotic labs have not only been used in engineering education but in other disciplines, across the curricula. Beginning as early as 1999 [3], reports have appeared on using robotics in teaching science courses. With the growing popularity of robotic devices in general, this has been made a trend in education, including some specific disciplines, such as computer science [4], spreading the ideas across the curriculum [5], with most recent papers reporting on deeper involvement of robotics across broad range of sciences [6].

However, in software engineering education, there have been only a few attempts of merging robotics and software engineering courses. Most notably, Roy et al. reported on their experiences of teaching software development in a robotics project course [7]. They described a curriculum for a two-semester course sequence in hardware-software development, involving a full development cycle for an autonomous mobile robot. They share their insights on teaching large-scale system development, with emphasis on software-intensive projects.

Christensen developed and taught a course on Software Engineering in Robotics [8], with a basic objective to introduce students to the fundamental issues of software engineering and how it applies to robotics. The software engineering part of the course covers the design, analysis and synthesis of systems for real-world operation, with presentation of some basic methods of software engineering, their application in robotics and the use of state-of-the-art software systems to implement these methods.

In contrast to the use of software engineering principles in robotics courses, described above, Göbel et al. report on using robotics in the software engineering program [9]. Their initial idea of using rather simplistic Lego Mindstorms robots soon had to be expanded to a simulation environment, since the student population got more interested in the subject matter and more sophisticated, requiring full-scale development and testing. The major observation from this project is that keeping focus on software, due to simulations, will let the students learn the algorithmic part of robotics, as opposed to dealing with the robot mechanics and electronics. The same observation has been independently confirmed by the current authors [10].

In this view, the objective of the current work is two-fold: first, to initiate a long-term project on incorporating robotics into software engineering curriculum, and second, being consistent with the evolution of the software engineering program at FGCU, involving online labs [11], to offer web-access to the robotic devices in the curriculum. It must be noticed that the second objective is much broader than in traditional online labs. It involves more than just remote access to control robotic devices, where students explore the opportunity to use robots remotely. Software engineering labs must be *invasive*, that is, the students have to be able to not only operate the devices remotely to conduct experiments, but also develop and test software remotely using these devices. This requirement is a significant game changer for online labs.

The objective of this paper is to present our initial experiences of designing an online robotics lab for use in the software engineering program. The rest of the paper is organized as follows. Section 2 presents the objectives of the lab, which is followed by the lab description in Section 3 and conclusion in Section 4.

2. PROJECT OBJECTIVES

2.1. Lab Background

Software Engineering and Robotics Lab developed at FGCU serves the purpose of supporting higher level courses in the curriculum by hands-on exercises offering students practical experience in software development for embedded real-time systems. With the ubiquitous use of the Internet and gradual expansion of the program towards offering courses via the web, a need arose to develop online labs.

Online engineering labs are relatively advanced and used worldwide [12], but the difficulty with online software engineering labs is that they must offer the students not only an opportunity to do remote experiments, just like in other science and engineering disciplines, but also a means to allow the modification of software running on a remote device. Software engineering labs must allow the students remotely modify, debug and update the software controlling the operation of a device. This is an essential requirement for such labs. Such lab has been developed and is in relatively uninterrupted operation since 2009 [13,14].

There are multiple issues with the development and operation of these labs, which can be categorized into three groups: pedagogical, technical and administrative. All of them, especially the pedagogy, the most important group of issues from the perspective of the learning processes, have been described previously [11]. The current paper addresses only technical issues related to the robotics part of the lab.

2.2. Robotics Lab Technical Objectives

Online robotic labs in a software engineering program should allow the students to remotely develop software for robotic devices, using Internet access for program upload and testing. A generic architecture of such lab is presented in Fig. 1. There are several technical issues, which need to be addressed for proper lab development and implementation.

First, and most importantly, proper Internet protocols need to be addressed for secure software, command and data transfers between the clients and the servers. Next, robotic devices equipped with software managing respective protocols need to be selected, for proper lab operation. Another important and decisive issue related the selection of robotic devices is their suitability with respect to pedagogy.

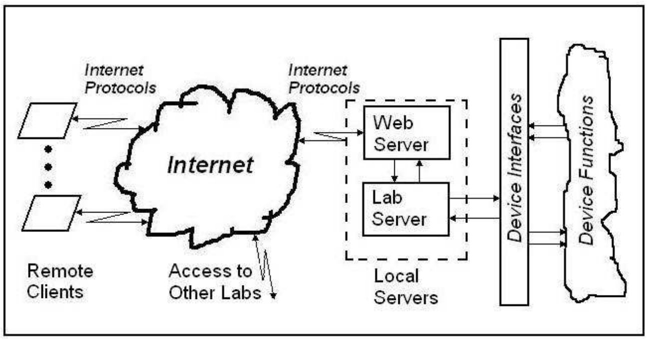

Fig. 1. Architectural components of a remote lab

Two remaining issues include the setup of servers, separated into a lab server, providing access to the devices, and web server, interfacing the lab to the external world. Finally, developing client software to access the devices has to be also taken into consideration, to cover a whole array of clients, from mobile phones to desktops.

3. LAB DESCRIPTION

3.1. Overview

Currently, there are around a dozen types of robotic devices acquired for the lab, from the simplest Lego Mindstorms NXT and EV3 to the sophisticated humanoid NAO robot and VEX and CoroWare vehicles. Keeping the diversity of robotic devices was one of the goals in their acquisition, to offer the students a broad range of option in using different technologies and learning related software development practices, at different levels of sophistication.

The list of available devices and their parameters is presented in Table 1. From the robotics perspective, robots can be described using a number of parameters, such a degrees-of-freedom of a robotic arm, etc. From the programming or software engineering perspective, however, what is important are computer related characteristics, such as the processor type, hardware interfaces, operating systems and programming languages available. Therefore, only these characteristics are included in the table, which shows the hardware features, software related features and lists vendors of each device.

Table 1. List of robotic devices in the lab

Robotic Device	CPU	Interfaces	Sensors	OS	Language	Protocols
CoroWare CoroBot[1]	Atom x86	WiFi	Kinect, mic	Windows 7	C#	http
VEX Clawbot[2]	ARM Cortex M3	usb, serial, VEXnet	gyro, IR tilt, ultrasd	none	Java, C, C++	ssh, telnet, rtsp
NAO Humanoid[3]	Atom x86	usb, WiFi, Ethernet	sonar, camera, mic ++	Gentoo Linux	C++, Python	http, ssh, NAOqi
Anybots QB Robot[4]	Intel Core 2 Duo	WiFi	camera, mic, distance	Free BSD	C/C++	http
AL5D Robotic Arm[5]	ARM Cortex A8	usb, WiFi, Ethernet, serial	camera, gyro, flex	Arch Linux	Javascript	http. ssh, tcp, udp
Lego Mindstorms:[4] NXT and EV3	Atmel (NXT) TI ARM (EV3)	Both have: usb, WiFi, Bluetooth	touch, color, gyro, IR RFID	none	G native, C, Java	http, tcp, udp
Parallax Boe Bot[7]	ATmega 328 Risc	usb, gpio, bluetooth	laser distance	none	C	Bluetooth
Hoverfly Copter[7] (one of 2 cpu's)	ARM Cortex A8	usb, WiFi, gpio, Ethernet, hdmi,	GPS & distance	Debian Linux	Python, Bonescrpt	http, ssh

Vendors:
1. CoroWare; http://robotics.coroware.com/corobot-robots
2. VEX Robotics; http://www.vexrobotics.com/vex
3. Aldebaran Robotics; http://www.aldebaran-robotics.com
4. Anybots; https://www.anybots.com
5. LynxMotion; http://www.lynxmotion.com/
6. Lego; http://mindstorms.lego.com/
7. Parallax; http://www.parallax.com/

Based on the device characteristics, the development of network connectivity for each robot has been accomplished as a project in a senior software engineering class. Student teams were assigned development tasks with full software engineering lifecycle, including requirements specification, design, implementation and testing. The next subsection describes some of these projects and their results.

3.2. Projects with Network Access

The essential component of each team project is to make a specific robotic device accessible over the Internet for programming. Selected projects are presented below.

CoroWare and VEX Robotic Vehicles. Both projects aim at making the wheeled robotic vehicle fully controllable over the Internet, with a capability to develop and modify the control software remotely. In the CoroWare project, the server application runs on the CoroBot onboard computer, which communicates with the Kinect sensor mounted on top of the vehicle, along with the robotic arm (Fig. 2).

The user interacts with the client side web browser, which sends and receives data to and from the CoroBot robot. These data are networked via the HTTP and TCP protocols from the client to the CoroBot server, and can represent a range of values including login information, commands, and C# source files.

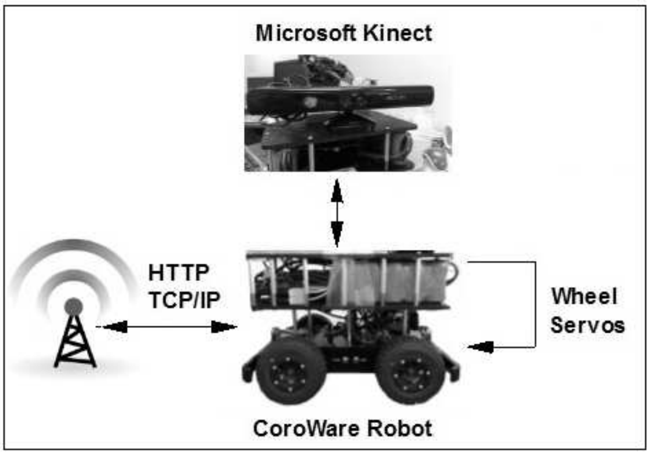

Fig. 2. CoroWare CoroBot with Kinect and Internet access

The VEX robotic vehicle follows the same principle of operation. The user can open the website in a browser and logon securely to the server, which then provides access to the control panel. From there the user can manually control the vehicle, as well as upload and update newly designed code. The user can also watch the behavior of the device via live video stream embedded within the web access panel. To develop software, the student must understand the operation of network, including HTTP, TCP and RTSP protocols.

Humanoid Robots. The humanoid projects aim at the same goals as the remote vehicle projects, described above. The NAO project includes live video stream fed to the client by the robot. Underlying is the communication between the client and the robot, with the client making calls to the robot and the robot returning data through a network connection.

A detailed diagram of the client/server interaction, with API function calls, is shown in Fig. 3. Software development and update is simpler in this case, because an SSH protocol can be used to connect to the NAO's onboard computer.

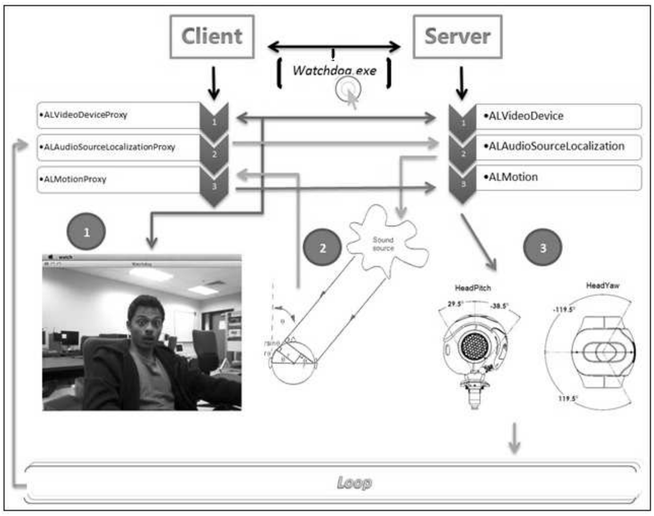

Fig. 3. Interaction of the client and NAO robot server

While our version of the NAO robot does not move, the Anybots QB robot's main feature, in addition to vision and speech, is movement. Due to vendor's restrictions, robot's internal software cannot be changed or replaced, so the project relies on developing Android application to operate the robot remotely via vendor's server access.

Other Robots. Limited space does not allow for presentation of other projects listed in Table 1, but the principle remains the same: providing web access to robotic devices, not only for control but also for remote software development and testing. Certainly, each device has its own peculiarities and poses some hurdles in reaching this goal.

For example, Lego Mindstorms devices are not designed for web access, which causes certain implementation problems. In addition, networking the Boe Bot robot equipped with Arduino board is difficult due to Arduino's limited capabilities. Finally, networking LynxMotion device had to be redesigned from Windows Embedded to Linux, due to better stability of Linux software releases. All these problems are very interesting from a learning perspective and add to the attractiveness of the project courses, letting students apply problem solving and critical thinking to produce unique designs.

3.3. Project Status

The initial objective of this work, offer web access to the robotic devices in the software engineering curriculum, has been partially achieved. Eight web-access robotics projects have been initiated and online access to each of these devices has been provided on an experimental basis. This completes the development phase. The next step is to actually use the robots, planned for two courses in the Fall 2014 semester: CNT 4104 Software Project in Computer Networks and CEN 4065 Software Architecture and Design.

There are certainly multiple technical problems that have not been adequately addressed, yet. One such crucial issue in invasive labs like this one, is the potential that an experiment may override the system software and even cause damage to the robot. Another non-trivial issue is whether after a safe but unsuccessful experiment the robotic device would return to a known state to be ready for the next experiment? Answers to these questions need to be addressed in the next stage of development of the lab.

4. CONCLUSION

Software engineering education for the modern world requires students to do lab work that would let them acquire skills, which are meaningful in the job market. In the age of the Internet, when courses and labs are more and more likely to be offered online, it is essential to build laboratories that follow this trend. In this project, a number of robotic devices have been connected to the Internet, with the goal to use them in the software engineering curriculum.

In addition to significant technical problems, such as protecting the robot hardware and system software from unintentional use, ensuring uninterrupted network and port access, projects like this face multiple organizational challenges, among them: faculty motivation and training, equipment maintenance and fees, tuition payment, and lab sustainability. To resolve all of them a more comprehensive approach is needed involving college administration.

The next immediate step in the project is to offer a software engineering course that will incorporate labs with Internet access and use these robotic devices for software development. A longer term goal is to develop a low level robotics course accessible to students in any field in Science, Technology, Engineering and Math (STEM), and to provide online access to use the robots for pursuing lab projects specific to these fields [15]. The implementation of this goal would require broadening the scope of the project to interdisciplinary collaborations.

5. ACKNOWLEDGEMENTS

This work has been supported by a grant from NASA through University of Central Florida's NASA-Florida Space Grant Consortium (UCF-FSGC 66016015). Partial support has been provided by the National Science Foundation (Award No. DUE-1129437), and U.S. Small Business Administration (SBAHQ-10-I-0250). Views and findings expressed herein are those of the authors and not necessarily those of the funding agencies.

Students of the FGCU computer science and software engineering programs are gratefully acknowledged for their contributions to the development of the lab.

6. REFERENCES

1. Computing Community Consortium, A Roadmap for US Robotics: From Internet to Robotics. May 21, 2009.
2. Granosik G.: Robotics in Education, J. of Automation, Mobile Robotics & Intelligent Systems, Vol. 8(1), 2014.
3. Beer R.D., Chiel H.J., Drushel R.F.: Using Autonomous Robotics to Teach Science and Engineering, Comm. of the ACM, Vol. 42, No. 6, pp. 85-92, June 1999.
4. Fagin B.S., Merkle L.: Quantitative Analysis of the Effects of Robots on Introductory Computer Science Education, ACM Journal of Educational Resources in Computing, Vol. 2, No. 4, pp. 1-18, December 2002.
5. Sklar E. et al.: Robotics across the Curriculum, Proc. AAAI/SSS Workshop Semantic Scientific Knowledge Integration, Stanford, Calif., March 26-28, 2008.
6. Datteri E. et al.: Learning to Explain: The Role of Educational Robots in Science, Themes in Science & Technology Education, Vol. 6, No. 1, pp. 29-38, 2013.
7. Roy N. et al.: The Experience of Teaching Software Development in a Robotics Project Course. Proc. 3rd Int'l CDIO Conf., Cambridge, Mass., June 11-14, 2007.
8. Christensen H.I.: Software Engineering in Robotics, College of Computing, Georgia Tech, Atlanta, Georgia, 2010. URL: http://www.cc.gatech.edu/~hic/8803-SER-10/Welcome.html
9. Göbel S., Jubeh R., Raesch S.L.: A Robotics Environment for Software Engineering Courses. Proc. 25th AAAI Conf, on Artificial Intelligence, San Francisco, Calif., August 7–11, 2011, pp. 1874-1878.
10. Gonzalez F., Zalewski J.: A Software-based Robotic Vision Simulator in Teaching Introductory Robotics, Proc. ECT2014, 9th Int'l Conf. on Electrical & Control Technologies, Kaunas, Lithuania, May 8-9, 2014.
11. Zalewski J.: Lab-by-Wire: Fully Web-based Hands-on Embedded Systems Laboratory, Proc. EDUCON 2013, IEEE Global Engineering Education Conference, Berlin, Germany, March 13-15, 2013, pp. 928-933.
12. Zubia J.G., Alves G.R. (Eds.), Using Remote Labs in Education. Bilbao: University of Deusto, 2011.
13. Daboin C., Zalewski J.: Lab Station for Remote Measurement and Control in Teaching Real-Time Embedded Systems and Software Engineering, Proc. 30th IFAC Workshop on Real-Time Programming, Mragowo, Poland, October 10-12, 2009, pp. 43-48.
14. Zalewski J.: A Comprehensive Embedded Systems Lab for Teaching Web-based Remote Software Development, Proc. CSEET 2010, 23rd IEEE Conf. on Software Engineering Education and Training, Pittsburgh, Penn., March 9-12, 2010, pp. 113-120.
15. Zalewski J., Gonzalez F.: Building an Undergraduate Robotics Lab Serving the STEM Curriculum, Proc. EDUCON 2014, IEEE Global Engineering Education Conference, Istanbul, Turkey, April 3-5, 2014.

ZDALNE LABORATORIA ROBOTYKI W ZAJĘCIACH INŻYNIERII OPROGRAMOWANIA

W ciągu ostatnich kilku lat, na Wydziale Inżynierii Florida Gulf Coast University, powstało bardzo złożone laboratorium inżynierii oprogramowania przenaczone do zajęć z systemów wbudowanych i innych kursów uwzględniających projekty zespołowe. Opracowane zostały odpowiednie moduły nauczania, na tematy związane z bezpieczeństwem, systemami złożonymi, i dostępem przez Internet. Celem obecnego projektu jest utworzenie laboratorium specyficznego dla potrzeb robotyki, z dostępem przez Internet, które moża wykorzystywać w innych dziedzinach, a wszczególności w inżynierii oprogramowania. Proces tworzenia laboratorium opiera się na istniejących urządzeniach, oprogramowaniu i modułach nauczania, z roszerzeniem ich w kierunku zastosowania w kursach z innych dziedzin, poza inżynierią. Laboratorium takie będzie miało wpływ na postęp w nauczaniu w takich dyscyplinach, jak nauki ścisłe, technologia, inżynieria i matematyka. W artykule opisano proces tworzenia i użycia modułów z wykorzystaniem ośmiu urządzeń-robotów, począwszy od prostych robotów Lego, do skomplikowanych pojazdów sterowanych zdalnie przez Internet.

Słowa kluczowe: zdalne laboratoria, laboratoria robotyki, nauczanie inżynierii oprogramowania.