

USABILITY EVALUATION OF A CONTROL AND PROGRAMMING ENVIRONMENT FOR PROGRAMMING EDUCATION

Maísa S. dos S. Lopes¹, Jessica O. Brito¹, Roque M. P. Trindade¹, Alzira F. da Silva¹ and Antonio C. de C. Lima²

¹Department of Exact Sciences and Technology, State University of Southwest Bahia (UESB), Vitória da Conquista, Bahia

²Electrical Engineering Department, Federal University of Bahia, Salvador, Brazil

ABSTRACT

This paper presents an assessment of usability of Control and Programming Environment (CPE) of a remote mobile robot. The CPE is an educational environment focused on computer programming education that integrates a program development online tool with a remote lab. To evaluate system usability, empirical test was conducted with computer science students in order to identify the views of users on the system and get directions on how to improve the quality of interface use. The study used questionnaire and observation of the evaluator. The degree of users' satisfaction was measured by using a quantitative approach that establishes the average ranking for each question of the questionnaire. The results indicate that the system is simple, easy to use and suited to programming practices, however needed changes to make it more intuitive and efficient. The realization test of usability, even with a small sample user, is important to provide feedback on the system's user experience and help identify problems.

KEYWORDS

Usability testing, programming education, remote lab, online compiler, mobile robotics

1. INTRODUCTION

Learning to program is a difficult task, especially for beginners [1]. Evidence of that is the high failure rate in disciplines related to this subject [2]. As a way to enhance learning, Husain et al [3] and Wu et al [4] suggest an increase in hand-on practice activities. However, the intense practical work required in programming is only sustained if the student is properly motivated [5]. To help in programming education, it developed the Control and Programming Environment (CPE) as part of LARA (Remote Laboratory in Virtual Learning Environment) project that integrates remote laboratory of mobile robotics with online tool for development program.

Remote labs are hardware and software systems that allows drive and control real experiments via the Internet. They have successfully been used in various disciplines of science and engineering [6, 7]. As an educational tool, remote laboratories should enable students to conduct their activities and make all the observations necessary to achieve the learning objectives [8]. However, its application in the programming education is little explored, few remote labs provide resources to support the activities inherent in this discipline.

A programming environment must allow access to tools that provide the building and running programs [9]. The remote laboratories described in [10, 11, 12, 13, 14, 15] allow only run programs. An exception is the SyRoTeck laboratory [16] allows the development and test programs using the NetBeans Integrated Development Environment (IDE) installed on the user's machine. However, for introductory courses simple programming tools should be used so that students do not spend too much time learning tools [9]. Therefore, the interaction with the software should be natural and intuitive providing good usability [17].

According to ISO 9241-11 [18], usability is "the extent to which a product can be used by specified users to achieve specific goals with effectiveness, efficiency and satisfaction in a specified context of use". Usability is a quality attribute related to ease of software use [19]. It refers to the capacity for learning and memorizing software, use efficiency, propensity degree to error and user satisfaction [20].

Usability evaluation is an important step in the development of software systems, especially for e-learning environments, it is a way to ensure that the system is adapted to the users, their tasks and that there are no negative results of its use [21]. To evaluate the usability of a software an empirical method or an inspection method can be used. In empirical methods, real end-users use the software (or a prototype) to perform a set of predefined tasks while the tester records the results for later analysis. In this case the evaluation of usability should happen in the last stages of the development process as a full or partial implementation of the software is required. The inspection methods do not require the participation of actual end users, the inspection is carried out by expert assessors verifying compliance of software artifacts with a set of usability guidelines [22].

The CPE is designed to allow programming students to perform practical activities in a simple environment, intuitive and easy to use. To ensure the quality of the proposed solution, a usability test with beginning students of the Computer Science course was held to know their opinion on system and identify improvements in their quality of interface usage.

The rest of this paper is organized as follows. Section 2 provides an overview of the system and its main components. Section 3 describes the usability test applied to evaluate the CPE's user interface. Section 4 presents some discussions about the results of the usability testing and, finally, in Section 5 we present our conclusions.

2. CONTROL AND PROGRAMMING ENVIRONMENT (CPE)

Remote laboratories use real components or instrumentation in a different location from where they are being manipulated. This allows them to be applied both in the classroom teaching and in distance education. The user accesses and controls the computer lab and then manipulates the equipment, makes observations, tests the conditions and collects experimental data. The installation of video cameras in the laboratory allows the user accompany real-time execution of the experiment. Modern remote laboratories use architecture based on the client-server paradigm and have a typical set of components: the controlled object (experiment, instrument or experimental model); equipment and instrument devices that allow data acquisition and control; lab server computer that provides control and monitoring of the experiment; the server that connects via the Internet between remote users and the laboratory server; the web camera server; and a client application that allows the user to access the experiment through a web browser [23,7].

The LARA CPE is a remote laboratory of mobile robotics integrated into a programming environment. Its architecture is shown in Figure 1. The robot is a line follower and rescue car based on the rules of the Robotics Brazilian Olympics (OBR) in the practice mode [24]. Communication between the robot and the laboratory server is performed via wireless network. The lab server is implemented in Java and run on a CubieBoard minicomputer. It is responsible for compiling and sending code for the robot, as well as opening and closing the serial communication. The LARA server is responsible for the connection between the user and the laboratory. The communication with the laboratory server is made through socket. For the development of LARA server were used: Apache web server, PHP5, HTML5, CSS3, MySQL, JavaScript and NodeJS. An IP camera is responsible for transmitting the images to the user in real time, for this VLC Media Player software is used.

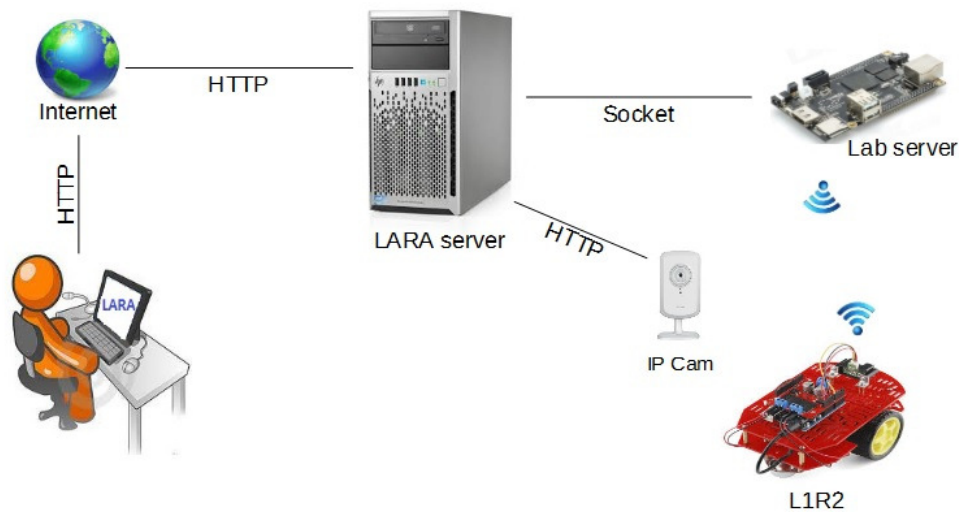


Figure 1. CPE architecture

CPE allows the user to access and control a mobile robot using a web browser. To access the robot, the user must login the system and make a reservation. The reserve consists of scheduling date and time of the experiment session. An experiment session is an interval of time allotted for the user to use the robot. The user starts the session when accessing the remote laboratory on time and reserved. During an experiment session the user can create a new code, open the sample code, compile a code, send a code to the robot, open a code that is on your computer and save a code on your computer. In addition, the user can watch the performance of the robot through the camera, receive feedback compiling via the console, send data to the robot through the serial, see which people are online in the system and talk to them by chat, follow the countdown of the session time and leave the remote lab.

Users access the features of the CPE through the graphical user interface (GUI) shown in Figure 2. GUI is divided in the following areas: 1 – toolbar, 2 – code editor, 3 – courseware, 4 – code examples, 5 – camera view, 6 – console that displays compilation of information, 7 – serial port and 8 – chat. The toolbar allows you to create a new code, compile a code, send a code to robot, save a code, open a source, open a serial port, open / close a camera, open a chat, check people online, follow the session timer and leave the environment.

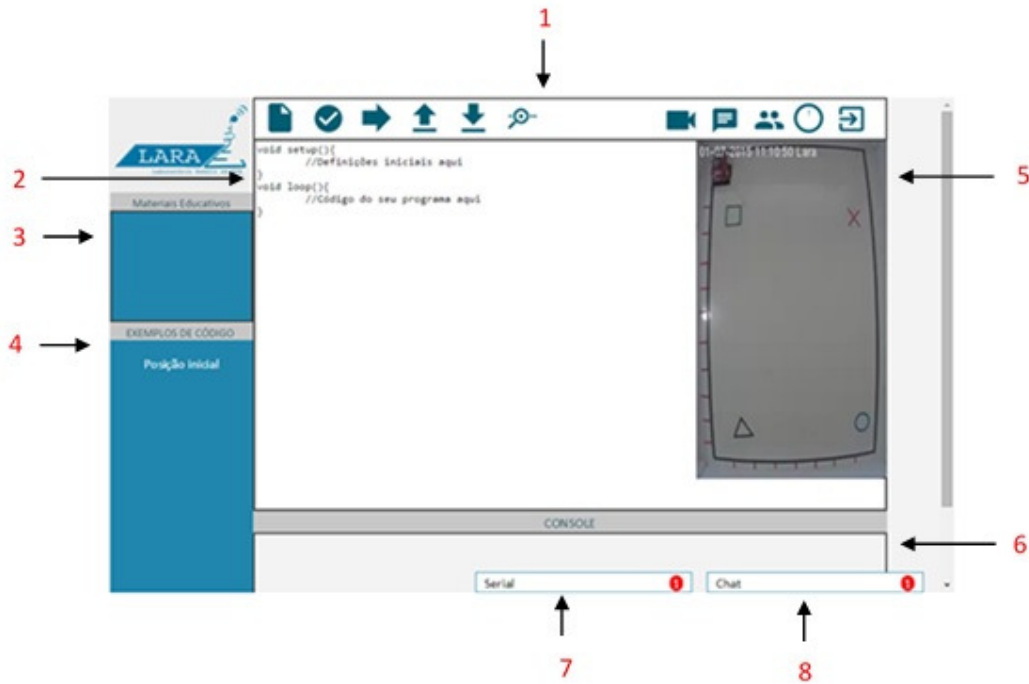


Figure 2. CPE user interface

3. USABILITY TESTING

The methodology used to develop the system was an adaptation of XP (eXtreme Programming) that in addition to software testing, predict the system usability testing. Other steps proposed by this methodology are outside our work scope and will not be described here.

The CPE usability testing sought primarily to assess the user's performance with the software, the goal was not to reach statistically valid results, but to have directions on how to improve the quality of the interface usage [25]. These tests were performed in the laboratory where the evaluator has greater control over the environment and user's activities.

The CPE usability testing process follows these steps: determining the objective of the evaluation, task selection, selection of participants users, preparing the material for the test, test execution and results analysis.

The objectives of the CPE usability testing were: Identify the views of users on environmental interface and its features and verify the effectiveness of the system to control the robotic device. Because it is a virtual learning environment and focusing on teaching programming language, the CPE must be accessible regardless of the degree of knowledge in virtual environments a user may have.

The selected tasks were: open the code file created in an earlier practice session, read the file instructions and check users' understanding, transfer the code for the robot, verify that the robot executes instructions in accordance with the code, and send commands to the robot via a serial port.

Following the recommendation in Nielsen [26] and Turner et al [27], we selected five users among UESB undergraduates of the Computer Science Course, who attended the first semester and were more than eighteen years old.

The test was conducted in the course lab and the following resources were used: a computer with Internet access and running logging program, an evaluator test script, tasks and an assessment questionnaire for the user, a stopwatch, a recorder, a pencil and a pen. After testing, each user answered questionnaire.

The metrics used in the preparation of the questionnaires and accompaniment script and observations of the evaluator were: subjective satisfaction, graphic layout, navigability, terminology, task execution time, task completion rate.

The first part of the questionnaire consists of five multiple-choice questions that aim to know the user's profile for the use of digital resources (beginner, intermediate, advanced), the use of virtual learning environments, age, programming knowledge and experience with robotics.

The second part of the questionnaire was used to obtain users' opinion about the CPE. It consists of twelve questions, with the first ten multiple-choice and two open ones. In the multiple-choice questions, the user specified the level of agreement with the question using a Likert scale of five points [28] (1: Strongly Disagree; 2: Disagree; 3: Neutral; 4: Agree; 5: Strongly Agree). The questions were:

1. Were you satisfied regarding the use of the system?
2. Were the layouts of the screens useful to the interaction in the system?
3. Were the information arranged in an organized manner?
4. Did the messages that appear on the screens help you to complete tasks?
5. Were the instructions for commands or functions objective unambiguous?
6. Were the instructions for error correction objective unambiguous?
7. Is the system easy to return to a previous state?
8. Is the system easy to learn?
9. Were the command names and uses easy to remember?
10. Does the system have flaws?
11. What is your impression on the control module screen for robotic devices?
12. What would you change in the interface?

RESULTS AND DISCUSSION

All users who participated in the test completed tasks and answered all questions. They were between eighteen and twenty-five years old, had never used a virtual learning environment, considered themselves beginners in programming and had no experience with robotics. Two students considered themselves intermediate in the use of digital resources and the rest of them considered themselves as beginners. Table 1 shows the responses of users to the second part of the questionnaire.

Table 1. Users' opinion about the CPE

Question	User 1	User 2	User 3	User 4	User 5
1	5	5	4	4	4
2	4	5	4	5	4
3	4	4	4	4	5
4	5	4	4	4	4
5	4	5	4	3	4
6	4	5	4	4	4
7	4	5	4	5	3
8	5	5	4	4	4
9	4	5	4	5	5
10	2	2	2	3	2
11	They are easy to display and interaction, but the size of the screen that shows the robot could be expanded	The screens were of great help to that there was a better experience than was being proposed.	First access, the control module screen looks very accessible to any user with little knowledge in the area.	I noted that it will be very interesting because it will give the user the opportunity to see your code work in practice without even being close.	Easy to interact, satisfactory.
12	The system takes to respond to commands would be interesting to add something that increases the speed of response to commands	The interface has everything you need and is easy to understand for those who are using, so would not change anything.	Apparently nothing.	Some buttons could be more objective.	I found the interface satisfactory, so I did not change anything.

The degree of users' satisfaction who tested the CPE was measured by using a quantitative approach that establishes the average ranking for each objective question of the second part of the questionnaire. The average ranking (R) was obtained by calculating the weighted average for each question, based on the frequency of responses as shown in Equation 1.

$$R = \sum (v_i * f_i) / 5 \quad (1)$$

Where:

- R – average ranking
- v_i – answer value i
- f_i – frequency response i

The R values greater than 3 are considered concordant and R values less than 3, as discordant. The result is shown in Figure 3.

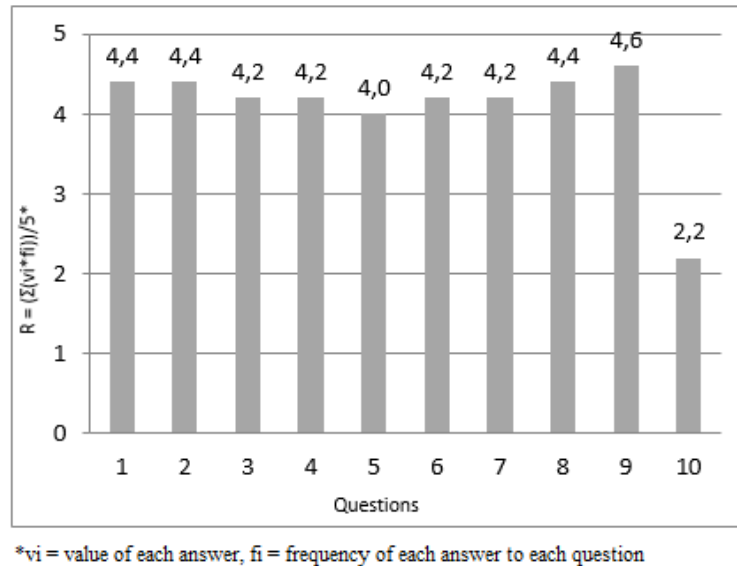


Figure 3. Average ranking of CPE satisfaction

Questions 1-9 had a mean ranking greater or equal to 4, which means that users were satisfied with the use of CPE. The layouts of screens help in the interaction with the system, information and messages are helpful and are organized, the instructions are objective, the system is easy to learn and commands are easy to remember. Question 10 obtained ranking equal to 2.2, it shows that the system has a few flaws.

Although the general evaluation of the system has been very good, open answers and observation of the evaluator indicated some aspects that needed improvement as shown in Table 2. Upon entering the environment, the user has no information on where it is. Users had a little trouble finding the button to compile the code. The resources to return the system to its previous state needed to be expanded. The sample code is not easily found. The system presented a failure because, while compiling the code, warning messages appeared on the console, but the code was written correctly. Users had difficulty finding the serial communication window and punctuated the need to improve system response speed.

Table 2. Usability flaws detected

Flaw	Description
Graphic layout	Page title
Terminology	Label the buttons
Navigability	Find the sample code and find serial
Error	Incorrect alert message
Task execution time	Delays to show the execution of the robot

Based on these indications, the following changes to the system were suggested:

- Set the environment name in the top of the page.
- Change the text of the labels command buttons to get more goals.
- To facilitate the return of the system to its previous state: put the robot's back button in the toolbar to starting position and provide the option to undo (ctrl + z) in the code editor.
- To facilitate the location of examples, to create the "Examples" button in the toolbar.
- The alert message from the system is not an error, but can cause frustration on the user. This problem was corrected in the laboratory server.
- To make serial communication window more visible, to put it next to the console tab.
- To increase system response speed, to improve configuration LARA server and connect it directly to the central switch of UESB network. Despite these measures, it is important to note that at a speed web system also depends on the user's network latency.
- As the CPE is used within the Moodle1 learning environment, it is not necessary to maintain the area of teaching materials.

Figure 4 shows the proposed new CPE interface with the changes described.

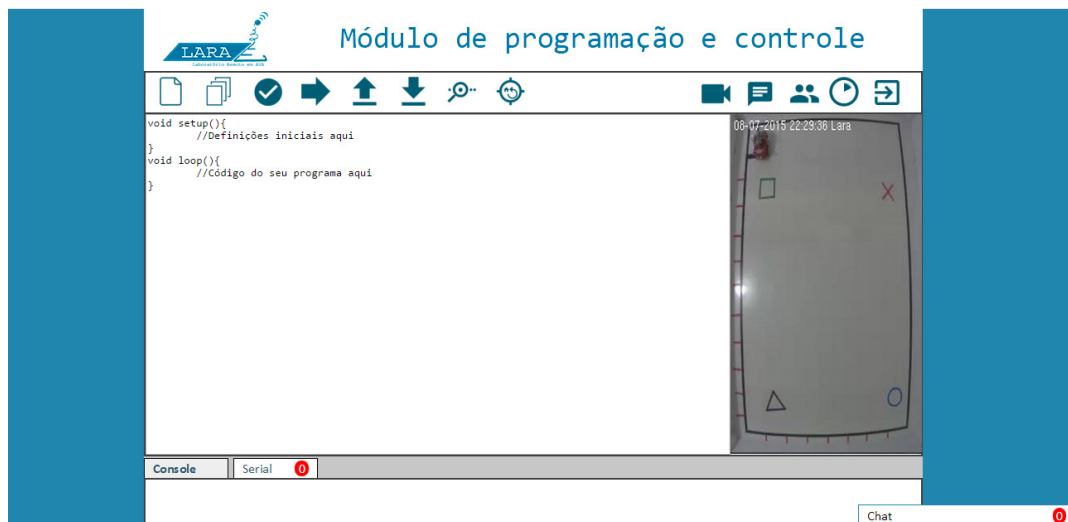


Figure 4. New CPE interface

5. CONCLUSION

Learning to program is a difficult process that requires a high load of practical activities. The CPE LARA was created to help in this process. It is a web system designed for teaching programming that incorporates in a single environment basic functions of IDE to access and manipulate a mobile robot. It allows developing applications for a real robot and view its execution. The system usability test was performed by five beginner students of computer science in order to verify the remote robot programming effectiveness and efficiency and possible improvements in

¹ <https://moodle.org/>

the interface. The results showed that users consider the CPE interface simple and easy to use and even with a small sample various using problems were identified.

The realization of empirical usability test was essential to give the CPE development team a feedback from users about the use experience of the system, to point improvement to make it more intuitive and efficient and show that it is possible to use it in computer programming education, in addition to showing the importance of developing educational software focused on the user. Yet it was also observed that the use of inspection test during the phases of software analysis and design can detect and correct primary errors as the lack of page title.

Currently the system is being used by students of computer science (UESB), discipline Algorithm and Programming I. Future work includes assessing the performance of students using the CPE.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge Helber H. Marinho, João V. O. F. Silva, Matheus L. Viana and Rodrigo Lima for their contribution in the implementation of software; and thank Iago Gomes, Thomas J. S. Teixeira and Vinicius Brito for help in developing for L1R2.

REFERENCES

- [1] T. Wang, X. Su, P. Ma, Y. Wang, and K. Wang, "Ability-training-oriented automated assessment in introductory programming course," *Computers & Education*, vol. 56, no. 1, pp. 220–226, 2011.
- [2] T. Jenkins, "On the difficulty of learning to program." In: *Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences*, 2002, pp. 53-58.
- [3] M. Husain, N. Tarannum, and N. Patil, "Teaching programming course elective: A new teaching and learning experience," In: *MOOC Innovation and Technology in Education (MITE)*, IEEE International Conference, 2013. pp. 275-279.
- [4] H. T. Wu, P. C. Hsu, C. Y. Lee, H. J. Wang, and C. K. Sun, "The impact of supplementary hands-on practice on learning in introductory computer science course for freshmen," *Computers & Education*, vol. 70, pp. 1–8, 2014.
- [5] K. M. Y. Law, V. C. S. Lee, and Y. T. Yu, "Learning motivation in e-learning facilitated computer programming courses". *Computers & Education*, 55(1), 218–228, 2010.
- [6] J. Ma, and J. V. Nickerson, "Hands-On, Simulated, and Remote Laboratories: A Comparative Literature Review". *ACM Computing Surveys*, 38(3), 7, 2006.
- [7] M. Tawfik, E. Sancristobal, S. Martín, G. Díaz, J. Peire, and M. Castro, "Expanding the Boundaries of the Classroom - Implementation of Remote Laboratories for Industrial Electronics Disciplines," *IEEE Industrial Electronics Magazine*, vol. 7, no. 1, pp. 41–49, 2013.
- [8] M. Cooper, "Remote laboratories in teaching and learning – issues impinging on widespread adoption in science and engineering education." *International Journal of Online Engineering (iJOE)*, v. 1, n. 1, pp. 1-7, 2005.
- [9] A. Pears, S. Seidman, L. Malmi, and L. Mannila, "A survey of literature on the teaching of introductory programming". *ACM SIGCSE Bulletin*, vol. 39, no. 4, pp. 204–223, 2007.
- [10] C. Daboin, and J. Zalewski, "Lab Station for Remote Measurement and Control in Teaching Real-Time Embedded Systems and Software Engineering". In *Proc. 30th IFAC Workshop on Real-Time Programming*, Mragowo, Poland, 2008, pp. 43-48.
- [11] O. Dziabenko, J. García-zubia, and I. Angulo, "Time to play with a microcontroller managed mobile bot". In *Global Engineering Education Conference (EDUCON)*, 2012 IEEE, pp. 1–5.
- [12] S. Kodagoda, A. Alempijevic, S. Huang, M. de la Villefromoy, M. Diponio, and L. Cogar, "Moving away from simulations: Innovative assessment of Mechatronic subjects using remote laboratories".

2013 12th International Conference on Information Technology Based Higher Education and Training (ITHET), 1–5.

- [13] D. Chaos, J. Chacón, J. A. Lopez-Orozco, and S. Dormido, “Virtual and Remote Robotic Laboratory Using EJS, MATLAB and LabVIEW” *Sensors*, vol. 13, n. 2, pp. 2595–2612, 2013.
- [14] C. M. Ionescu, E. Fabregas, S. M. Cristescu, S. Dormido, and R. Keyser de, “A Remote Laboratory as an Innovative Educational Tool for Practicing Control Engineering Concepts,” *IEEE Transactions on Education*, vol. 56, no. 4, pp. 436–442, 2013.
- [15] M. Casini, A. Garulli, A. Giannitrapani, and A. Vicino, “Remote pursuer-evader experiments with mobile robots in the Automatic Control Telelab.” In *10th IFAC Symposium on Advances in Control Education*, Sheffield, UK., 2013, pp. 66–71.
- [16] M. Kulich, J. Chudoba, K. Košnar, T. Krajník, J. Faigl, and P. Libor, “SyRoTek — Distance Teaching of Mobile Robotics,” *IEEE Transactions on Education*, vol. 56, no. 1, pp. 18–23, 2013.
- [17] M. F. Costabile, M. De Marsico, R. Lanzilotti, V. L. Plantamura, and T. Roselli, “On the Usability Evaluation of E-Learning Applications. In *System Sciences*”, 2005. HICSS '05. Proceedings of the 38th Annual Hawaii International Conference on. IEEE, pp. 1–10.
- [18] ISO 9241-11, 98: Ergonomic Requirements for Office work with Visual Display Terminals. Part 11: Guidance on Usability. ISO. 1998.
- [19] N. Juristo, A. M. Moreno, and M. I. Sanchez-Segura, Analysing the impact of usability on software design. *Journal of Systems and Software*, 80(9), 1506–1516. 2007.
- [20] J. Nielsen, “Usability 101: Introduction to usability”, 2003, Available: <http://www.nngroup.com/articles/usability-101-introduction-to-usability/>.
- [21] J. M. C. Bastien, “Usability testing: a review of some methodological and technical aspects of the method”. *International Journal of Medical Informatics*, 79(4), e18–e23. 2010.
- [22] A. Fernandez, E. Insfran, and S. Abrahão, “Usability evaluation methods for the web: A systematic mapping study”. *Information and Software Technology*, 53(8), 789–817. 2011.
- [23] L. Gomes, and S. Bogosyan, “Current Trends in Remote Laboratories”. *IEEE Transactions on Industrial Electronics*, 56(12), 4744–4756. 2009.
- [24] OBR. Olimpíada Brasileira de Robótica Regras Final Nacional – RoboCup Junior Rescue A - Modalidade Prática. 2014.
- [25] R. O. Prates, C. S. de Souza, and S. D. Barbosa, “Methods and tools: a method for evaluating the communicability of user interfaces”. In *Proc. of Interactions*, v. 7, no. 1, pp. 31–38, 2000.
- [26] J. Nielsen, “Why you only need to test with 5 users,” 2000, Available: <http://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>.
- [27] C. W. Turner, J. R. Lewis, and J. Nielsen, “Determining Usability Test Sample Size”. *International Encyclopedia of Ergonomics and Human Factors*, 3(2), 3084–3088. 2006.
- [28] G. Albaum, “The Likert scale revisited: An alternate version”. *Journal of the Market Research Society*, vol. 331, no. 2, pp. 1–12, 1997.

Authors

Maísa S. dos S. Lopes received degree in computer science from the Federal University of Viçosa (UFV), Minas Gerais, Brazil, in 1998, the M.Sc. degree from the Federal University of Pernambuco (UFPE), Recife, Brazil, in 2003, and is currently pursuing the Ph.D. degree in industrial engineering at the Federal University of Bahia (UFBA). Since 2004, she has been an assistant professor of computer science course at the State University of Southwest Bahia (UESB). Her current research interests include programming education, remote lab, robotic, Computer Supported Collaborative Learning (CSCL) and online education.

Jessica O. Brito received the degree in Computer Science from the State University of Southwest Bahia (UESB), Vitória da Conquista, Brazil, in 2015, and is currently pursuing the M.Sc. degree in Computer Science from Federal University of Espirito Santo (UFES). She is interested in the field of human-computer interaction.

Roque M. P. Trindade received the degree in Enabling Science in Mathematics from the State University of Southwest Bahia (UESB), Vitória da Conquista, Brazil, in 1994, the M.Sc. degree in computer science from the Federal University of Pernambuco (UFPE), Recife, Brazil, in 2002, and Ph.D. in electrical engineering from the Federal University of Rio Grande do Norte (UFRN), Natal, Brazil, in 2009. He is currently an adjunct professor at the UESB. He works in the computer science acting on the following topics: interval mathematics, signal processing, intelligent systems, control and automation.

Alzira F. da Silva received the degree in Enabling Science in Mathematics from the State University of Southwest Bahia (UESB), Vitória da Conquista, Brazil, in 1995, the M.Sc. degree in computer science from the Federal University of Pernambuco (UFPE), Recife, Brazil, in 2002, and Ph.D. in electrical engineering from the Federal University of Rio Grande do Norte (UFRN), Natal, Brazil, in 2009. She is currently an adjunct professor at the UESB. She has experience in computer science, acting on the following topics: Intelligent Agents, Educational Robotics, Computational Mathematics, Computers in Education.

Antonio C. de C. Lima received the B.Sc. degree in electrical engineering from the Federal University of Bahia (UFBA), Bahia, Brazil, in 1988, the M.Sc. degree from the State University of Campinas (UNICAMP), Campinas, Brazil, in 1990, and the Ph.D. degree in electronic engineering from the University of Kent, Canterbury, U.K., in 1994. He joined the UFBA Electrical Engineering Department in 1994 as a Professor of Applied Electromagnetism. In 2002, he completed his postdoctoral activities at McMaster University, where he developed research in the area of Wireless Communications. He is currently Professor at the Federal University of Bahia. He has experience in Electrical Engineering with emphasis on Applied Electromagnetism, Communication Systems and Digital Signal Processing, acting on the following topics: Embedded Systems, Digital Signal Processing, Artificial Neural Networks and Adaptive Equalization.