

Pedagogical architectures and web resources in the teaching-learning of programming

Lucinéia Barbosa da
Costa
UFES
cliklucineia@gmail.com

Orivaldo de Lira
Tavares
UFES
tavares@inf.ufes.br

Crediné Silva de
Menezes
UFRGS
credine@gmail.com

Rosane Aragón
UFRGS
rosane.aragon@gmail.com

ABSTRACT

This paper presents pedagogical architectures supported by resources from the web, specifically designed to aid the learning of computer programming. Based on Piaget's theory and research on the use of ICTs in education, this article presents research findings on the use of pedagogical approaches that include web resources to enhance the development of important skills to program computers.

Keywords

Pedagogical architecture, computer programming, teaching and learning, ICT, web.

1. INTRODUCTION

Each year, the use of digital resources is gaining more space in education. Reis and Godói [6] claim that there is no doubt that one of the most remarkable effects of the technology is to enable and facilitate greater communication between people. With the increasing use of technology, digital web resources begin to promote a paradigm shift from teacher-centered instruction to a more constructivist approach, focused on student and cooperation among the participants.

Donato and Guimarães [2] claimed that the incorporation of ICT use in education is an irreversible process, whereas information and communication technologies are part of everyday life. Based on the theory of Piaget [5] this paper presents pedagogical architectures for the teaching and learning of programming, with support of web resources, which are designed to improve the learning and stimulate the development of programming skills in beginners of undergraduate courses in computing.

The paper is organized as follows: section 2 discusses the difficulties of learning computer programming; Section 3 discusses the pedagogical architectures with web technology in support of learning programming, section 4 presents the research methodology used, the section 5 describes the web technologies to support the pedagogical approach used, section 6 brings the experiences and results and section 7 presents the conclusions and, at following, the references.

2. DIFFICULTIES IN PROGRAM

The difficulties in learning computer programming refers to the fact that the skill to program is associated with a high load of abstract concepts in addition to the characteristics of programming languages, increasingly sophisticated.

Gomes and Mendes [3] reported that the major mistakes made by students during the learning process are errors related to syntax

and semantics, difficulties in understanding the problems, design of algorithms and error detection in program logic.

Tavares et al [7] argue that the difficulties in learning programming are related to the development of various skills that are not addressed before the student enters the university, such as:

- domain of a methodology for problem solving;
- domain of a methodology for assessing the quality of programs produced;
- ability to solve problems cooperatively;
- domain of computational tools for the creation and evaluation of programs.

All these factors lead the student who has little or no prior knowledge on the subject to feel incapable of continuing in computing courses. When he/she needs for a personalized follow, the teacher is not always available to offer and nor traditional teaching methods have given support.

3. PEDAGOGICAL ARCHITECTURES WITH WEB TECHNOLOGY IN SUPPORT FOR LEARNING PROGRAMMING

Approaches using web technologies have gained increasing space, making it possible for students a more open and collaborative learning, with a focus more constructivist. In learning computer programming, the use of web technologies is crucial to support pedagogical approaches that enhance learning.

Carvalho et alli [1] define pedagogical architectures as learning structures from the confluence of different perspectives, such as: software, internet, artificial intelligence, distance education, conception of time and space.

Seeking to improve the teaching and learning of programming, researchers work with three dimensions: tools, strategies and strategies with tools. The use of a well-structured pedagogical architecture in the teaching and learning of computer programming represents a paradigm shift in knowledge construction. With it, the emphasis is on pedagogical approach in which digital and web resources are merely tools used to support the pedagogical approach.

4. METHODOLOGY

The survey was conducted with students from computer programming classes at a university in Brazil. We conducted a quantitative descriptive study, with the use of questionnaires,

analysis of reports of students about building of programs, monitoring class in problem solving and analysis of student behavior throughout the course. The pedagogical architecture used in this work consists of two essential components: a pedagogical approach and the web technological resources to support usage of the approach.

To develop the pedagogical approach used in the experiments, it was necessary to analyze the needs of each student and set goals, objectives and appropriate methodological procedures. This pedagogical approach aimed to promote development in students of the following skills: problem solving individually, solving problems in a cooperative way; arouse the interest of students in search of enlightening information of your doubts; socialization of discoveries; peer review work; learning by observing and comparing solutions.

During the research, it was suggested to the class to use the Method for Program Construction (MCP). The MCP is a method developed by Tavares et al [8]. It consists of six steps, in which students go through each one to build a program. Figure 01 shows the steps of the MCP.

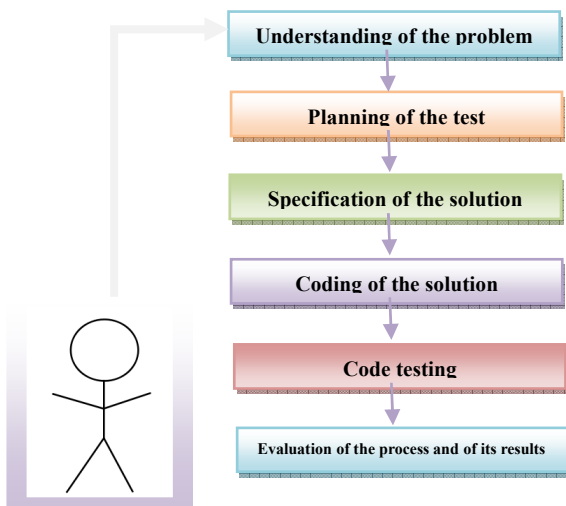


Figure 01 – Method for program construction – MCP

5. WEB TECHNOLOGIES AS SUPPORT FOR PEDAGOGICAL APPROACH

Education professionals seek to constantly create innovative pedagogical approaches that require new technological resources to support them.

Several technologies have been developed to support teaching and learning programming [4]. These technologies represent a valuable contribution to learning programming. However, there are few pedagogical approaches that use web resources to really enhance learning.

The following are the digital resources used to implement the pedagogical approach used in a class of programming offered to undergraduate students in computer science and computer engineering, from a university in Brazil. They are:

Wiki - this resource of asynchronous communication was used to create a space for publication of ideas, for cooperation among participants (students and teachers) and to analyze the relationships formed and its implications on learning.

Forum - this resource of asynchronous communication was used to allow joint reflections on the difficulties encountered by the students and on the chosen solutions to problems.

Room for interaction (chat) - this resource allows synchronous interactions between course participants. This resource was used for reflections on matters related to the activities presented to students.

Problem Solving – with this resource, the student accesses the problems posed by the teacher and develops solutions. The student chooses an exercise of a list of problems and using a report editor, reports the development of each solution.

Questionnaire - this resource of asynchronous interaction was used as online activity by students. Questionnaires were used to collect information from students about the issues presented in the classroom, in order to guide the pedagogical practices of the teacher.

Task - this resource of asynchronous interaction allows teachers to publish an activity on the environment and get the solutions of each student. After learning of the new task, the student develops the requested activity and sends it to the teacher through a web resource.

Games for logical reasoning - several games of logic were applied to put students in situations where they could exercise the logical thinking and the skills needed to solve complex problems.

Web - all digital resources cited were available on the web, so that students could develop their tasks within or outside of the classroom and interact with other participants of the class.

6. RESULTS

Participants were 30 students of Computer Science and Computer Engineering from a university in Brazil. The students were aged 20-30 years. The percentage of 87% of the sample was male and 13% female. Some students had previous knowledge in programming languages, as shown in Figure 01.

A total of seven lists of exercises were applied. All lists have the same goals, what differed was the degree of complexity. Table 01 presents the lists of exercises, the amounts of activities and their respective goals.

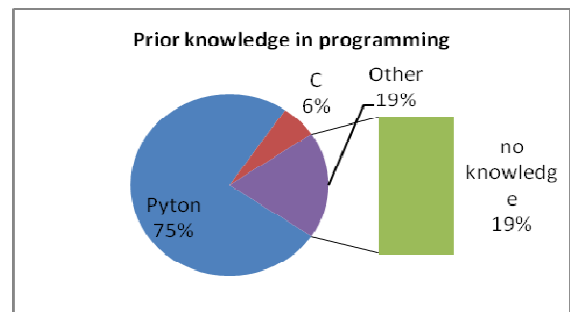


Figure 02 – Prior knowledge in programming

During the research, we check which lists of exercises in which students felt more difficulties. Table 02 presents the percentage of students who had difficulties in the development of the lists.

Table 01- Exercises

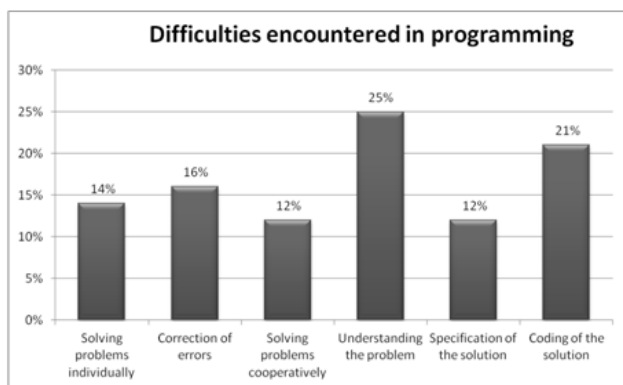
List exercises	Quantity exercises	Objectives
List 1	9	Analyze student involvement in activities; Identify the student's abilities to solve problems; Analyze the various forms of problem solving developed by students; Check the main difficulties obtained by students.
List 2	2	
List 3	2	
List 4	2	
List 5	1	
List 6	1	
List 7	1	

Table 02- Difficulties presented in the lists of exercises

List exercises	Presented difficulties	Not presented difficulties
First	56%	44%
Second	37%	63%
Third	33%	67%
Wednesday	37%	63%
Farm	31%	69%
Friday	44%	56%
Seventh	37%	63%

On a scale of 0% (zero) to 100% (one hundred), considering all the lists of exercises presented, the first list had the highest rate of problems: 56%. This index is partially justified by the fact that students were adapting the pedagogical architecture.

The survey also sought to identify the main difficulties encountered by students in the tasks. Figure 02 shows the percentage of the difficulties encountered by students during the resolution of the activities.



Graphic 02: Difficulties in developing programs

We build learning indicators that are specially used for this purpose. These indicators are:

Construction of solutions: the student is encouraged to build several different solutions to a single problem. This method encourages students to build many new solutions.

Independence in learning: independence in learning makes students look to solve problems independently, developing in him the creativity and safety.

Find ways to make the code easier: after encode a solution, a good developer search for new solutions even better: simpler and more efficient.

Correction and test of functions: not always created codes are correct. It is then that the student needs to rely on his plan to test the solution in order to identify bottlenecks and adjustments to be made.

Importance of considering various solutions: we learn more when we compare many possible solutions. The student should not settle with a single solution for each problem. He should consider various solutions, seeking the best among them.

Understanding of the problem before solving it: before attempting to solve any problem you need to understand it deeply, so that the chosen solution meets the specification of the problem.

Advantages of working collaboratively: complex problems require collaborative work. In the academic environment, collaborative work needs to be exercised in order to be learned how to perform it.

The motivation to develop programming projects: The student must be motivated to develop programming projects, even outside the school environment. This allows him to gain experience in building programs in the near future and become a good programmer.

Reflection on own solution: The student should reflect on solutions built and analyze in detail every developed solution.

During the development of this work, we realized the growing interest and participation of students in the classes of computer programming, culminating in increased socialization of knowledge and significant improvement in student performance.

7. FINAL CONSIDERATIONS

The learning process of computer programming increasingly should be studied in order to improve performance and student learning. Pedagogical approaches get priority with the emergence of new technologies of configuration, more flexible and fast, of digital resources that prioritize both usability as educational efficiency [9].

That reminds Reis et alli [6] when they say that, in a changing world, new challenges arise daily in the school environment. This paper presented the results of a research on pedagogical architectures for teaching and learning computer programming, supported by digital resources and web.

The main objective of this work was experiment with new pedagogical approaches supported by technology to improve learning about computer programming of freshmen of computer undergraduation courses.

The data analysis shows that the pedagogical approach with technological support used in this research contributed to student learning, as shown by the learning indicators used to measure the development of programming skills of the students. These indicators are also important for designing pedagogical architectures suitable for improving learning.

8. REFERENCES

[1] Carvalho, M.J.S.; Nevado, R. A.; Menezes C. S.; "Pedagogical Architectures for Distance Education: Concepts and Telematic Support";

- <http://peadsaoleopoldo.pbworks.com/f/arquiteturaspedagogicas5.pdf> available; played on 14/06/2010. Danquigny, T: Numérique, nouvelles générations et pratique d'études. In: World Wide Web, 2012. Lyon France.
- [2] Donato, A. F.; Guimarães, R. B. Without distance education: a challenge for virtual learning environments, p. 26. In: Information technology and communication (ICT) in developing Professionals Unified Health System (SUS) Collection Issues in Public Health. Volume 12, 2011.
- [3] Gomes, A; Henriques, Joan; Mendes, JA (2008). A proposal to help students with learning difficulties early computer programming, Coimbra 2008.
- [4] Gomes, A. and Mendes, A. (2000). Support for Learning Programming with Environment SICAS. Proceedings of the V Latin American Congress on Computers in Education, Viña del Mar, Chile
- [5] Piaget, J.; Intellectual Evolution from Adolescence to Adulthood; Source: Human Development, 15:1-12, 1972.
- [6] Reis, E.; Godói, KAA dialogicity tutoring in the distance: a contribution Freire. Proceedings of the VIII International Seminar Paul Freire. Recife, 2010.
- [7] Tavares, O. L.; Menezes, CS; Nevado, RA: Pedagogical architectures to support the process of teaching and learning of computer programming: In-FIE2012 Frontiers in education conference, 2012.
- [8] Tavares, O. L. , Santana, L. S. , Menezes, C. S. ; Nevado, R. A.: Architecture for pedagogical mediation of the teaching-learning computer programming. In: V Workshop of Pedagogic Architectures to Support Distance Education Internet-mediated, 2010, João Pessoa / PB. Annals of the Brazilian Symposium on Computing in Education: SBC, 2010. v. One. p. 1-10.
- [9] Vieira, Jr., R. R. M, Santos, OL; Rafalsky, JP; Bada, IN; Silva, HFA , Menezes, C. S. . Coordination in Collaborative Activities in Learning Environments - An Evaluation of the Implementation Architectures Pedagogical. RENOTE. New Technologies in Education Journal, vol. 9, p. 161-170, 2011.