

PBL in Programming Subjects at Engineering

A. J. Sierra, T. Ariza, F. J. Fernández

Abstract—This paper presents the PBL learning methodology applied to the subject of Programming in Engineering. We show two different perspectives, teacher and student. First, the point of view of the teacher considers the design of the project, and secondly student that considers its implementation.

Index Terms—Communication engineering education (Engineering education), Computer science education, Engineering students. Programming (Computers and information processing), Problem- and Project-based learning

I. INTRODUCTION

SINCE 2010 the Department of Telematics Engineering in the University of Seville, subjects are taught in the new DTTE (Degree in Telecommunication Technology Engineering), to converge with the EHEA (European Higher Education Area) recommendations. This is an ambitious and complex plan launched to promote European convergence in education. The foundation of EHEA is based on student's work and laboratories are essential to develop theoretical content and implement abilities in a larger scale work.

Programming content is taught in two subjects in the first course: Fundamental of Programming I (FPI) and Fundamental of Programming II (FPII). The purpose of these subjects is to establish the principles of computer programming.

FPI is a term core subject at the first-year of DTTE. This subject consists of 6 ECTS (3 ECTS credits are lectures and 3 ECTS credits are practical classes).

FPII is a term core subject at the first-year of DTTE. This subject consists of 6 ECTS (1.5 ECTS credits lectures and 4.5 ECTS credits are practical classes). This subject has a high practical content. The methodologies used for the acquisition of knowledge are the following: lectures, laboratory practices, and Active use of e-learning technologies.

Every student must complete a course project that will consist of developing an application, that consist of developing an application, comprising the steps of understanding the problem, designing of the program, coding it in C and subsequent testing. Furthermore, every student must complete another course project that will consist of developing an application, comprising the steps of understanding the problem, designing of the program, coding it in Java and subsequent testing.

Manuscript received Mars 30, 2013. This work was supported in part by the Department of Telematics Engineering in the School of Engineering at University of Seville.

J. Sierra, T. Ariza and F. J. Fernández are with the Department of Telematics Engineering in the School of Engineering at University of Seville, C/ Camino de los descubrimientos s/n, 41092, Seville, Spain (phone: 349-448-7384; fax: 349-448-7385; e-mail: {antonio, teresa, fjfj}@trajano.us.es).

Project and problem-based learning (PBL) are perhaps the most innovative instructional method conceived and implemented in education, and has been widely recognised as an active, collaborative, cumulative and integrative learning approach that engages learners, motivates team creativity and centers on practical education. It aims to enhance students' application of knowledge, problem solving skills, higher-order thinking, and self-directed learning skills.

This instructional method has been successfully applied in different educational disciplines. Hung[8] builds on the 3C3R problem design model, a 9-step problem design process. This model does not fit the design of projects in programming courses, due to the complexity associated with software project.

We propose a new model, based on the 9-step of the 3C3R model, to design the project for the programming in FPII.

This paper is organized as follows, first we present the PBL in programming labs, where we shows both, a review state in PBL, and the context in the programming's subject. Second, we present how to design the project. Then, we show the project implementation, and finally conclusions.

II. PROJECT-BASED LEARNING IN PROGRAMMING LABS

A. Review Stage in PBL

PBL was originally conceived and implemented in response to students' unsatisfactory clinical performance [3] resulting from the emphasis on memorization of fragmented biomedical knowledge in traditional health science education.

The widely adopted format of PBL was first developed in medical education at McMaster University in the 60s and 70s [3]. Since its first implementation, PBL has become a prominent instructional method in medical and health science education throughout the world.

With the positive results from implementing PBL in medical education, PBL has also been embraced by other disciplines in higher education, such as architecture [10], law schools [12], leadership education [5], nursing [4], and teacher education [9], science courses [1], biochemistry [11], calculus [14], chemistry [2], economics [7], geology [18], and psychology [13].

Hung[8] builds on the 3C3R problem design model, which is a systematic conceptual framework for guiding the design of effective and reliable problems for PBL. To help practitioners apply the 3C3R model, this model introduces a 9-step problem design process.

This model to design problems (in 9 steps 3c3r) proposed by Hung does not comply with design of projects in programming courses because of the complexity associated with software project. Our design model for the programming project is based on Hung. We have adapted the Hung's model to consider the specific characteristics of a software project.

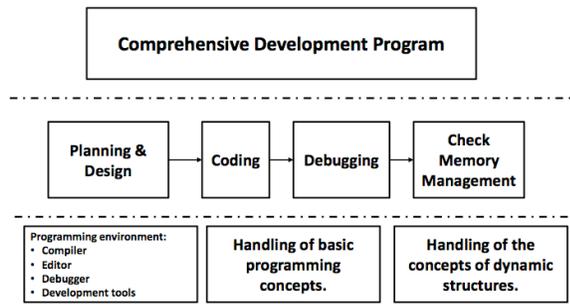


Fig. 1. Link content to project tasks

B. Context in the Programming's subject

The programming courses usually require practice and projects to assimilate knowledge by the student. Therefore, since many years, this course introduced the PBL model. The skill of this subject is taught through guided practice where the student has to deliver different types of exercises related. The practices involve training the skills required to develop a larger software project. A student must develop a software project that requires a series of tasks to perform (which has not faced ever). All possible implementations can be different and all potentially valid, but must meet certain requirements. Software engineering has made a series of development phases of a project. The proposed project develops the concepts and content of structured programming. The topic and structure must meet the following aspects:

- **Contextualized:** The topic must be in the telecommunication context.
- **Real-world:** The project should be a case taken from the real world.
- **Ill-structured:** The problem must be of adequate size so it can be broken down into parts where required design criteria. Typically, students in their implementation can be found with the following problems:
 - **Poor structuring.** There is usually a trend to approach the project as "a whole", coding is performed with a complete program without structure, without modules.
 - **Without running tests.** No validation is performed, or is performed testing the modules implemented, until the final stage of the project.

The teacher's job is to design a project ill-structured, appropriate to the student's knowledge, and to tutor in order to guide students in their development to achieve carry it out properly. The statement of the project should be complete, well specified, self-contained, real-world, drafting and design requires a series of steps very carefully developed to serve as a standard specification or project design. These design steps shown in the next section.

II. DESIGNING THE PROJECT

This section presents the steps required to design a project in a programming course.

Step 1: Set objectives and goals.

In this step the objectives and goals of the course are specified. Both the depth and breadth of content are set and also skills the student should acquire by developing the project. In FP2, the main goal is to learn the basics of programming in an

imperative language, such as acquire the necessary skills to manage dynamic structures.

Step 2: Link content to project tasks.

In this step how to get the learning objectives set in step 1 are decided. The objectives are broken down into tasks.

The knowledge goals set in step 1 will be achieved through the development of a software project, where the student must use basic concepts of structured programming and dynamic structures. This idea is shown in the following Fig. 1.

Paso 3: Specify the context.

In this step the following context are specified:

A. Real-world context:

This context refers to the general topic of the project. In fp2 is the context of telecommunications, since the subject belongs to the DTTE.

B. Work context:

This context refers to whether the work is individual or collective, the location where the project is developed and the platform that is used for communication between teacher and student.

- In the case of the subject of FP2,
- The work is performed individually.
- The physical placement where students perform the project is in the data center and classrooms electrified in the school. This development can also be done at home, using a similar environment.
- To communicate teacher and student, WebCT platform is used.

C. Programming Environment.

In this context the programming language, the tools used for development, and virtualized environments are specified.

In order to develop the project in fp2, the C programming language, the gcc compiler, emacs and gedit editors are used. Besides, a virtual machine environment for the development of the project is provided, so that a student can develop your project anywhere.

Step 4: Select/generate PBL problem.

Step 3 determines the context of the project, determining the theme, location and environment where the problem will be developed. In this step 4 a specific problem is selected for achieving the targets set in step 1.

We have been developed a list of project, within the topic of telecommunication engineering. These problems are:

1. NAT: Network address translation
2. Packet Forwarding System.
3. Calculating the shortest path in a packet network.

Step 5: Project affordance analysis and calibration.

This step is an analysis to determine whether the proposed project is affordable, and a calibration that allows adjusting the concepts and skills needed to develop the project. This task includes the following steps:

A. Understanding the actual operation of the system

proposed: The teacher develops the subject of the project to propose a particular situation. The functionality of the system must be defined to make it affordable for the student

B. General description of the project: Once the general topic of the project has been decided, a statement is described corresponding to the knowledge of the students who must

carry out the project. This general description should describe:

- Concepts used in solving the problem.
- Principles used in the solution of the problem.
- Procedures used in the solution of the problem.
- Factual information necessary to solve the problem.

C. Evaluate the student's knowledge to address the project.

Here, that the student has acquired the necessary knowledge is verified, otherwise it must be provided in the statement. That the student has the skills to develop the project should also be analyzed. In case of lack of skills, teachers should guide development in these skills. This represents a calibration of the concepts and skills of the student to adjust to the required skills to develop the project.

D. Detailed Description. At this point a detailed description of the inputs and outputs must be made as concretely as possible and real examples must be given. This detailed description should include the necessary documentation to guide students in those tasks that in the previous section has detected some lack.

E. Project implementation by teachers. At this point the proposed project is implemented by teachers to assess different aspects. These aspects are the difficulty of the project, the breadth and depth of content, errors in the statement. This implementation helps detect details that have not been specified in the first statement.

The result of this analysis is a statement of a project that is affordable for students. It also allows having a working implementation of the project.

Step 6: Correspondence analysis.

It is evaluated at this point if the proposed project includes all the knowledge and skills that have aimed at point 1.

Step 7: Documentation.

The documentation that the student must submit is determined in this step.

Step 8: Construction of a test suite. This step is to build a test suite that includes all possible cases in the project. This test suite allows checking the correct functionality of the project in all cases included in the statement.

In FP2, several configuration files and input files that are consistent with the statement are constructed. They define the situation in which the system generates some kind of error.

Step 9: Enabling a mechanism for discussion.

It is necessary to determine a form of communication between teacher-student and student-student to resolve any questions about the statement, design, implementation and testing of the project.

En fp2, a través de WebCT un foro específico para el proyecto es habilitado.

Step 10: Establishment of a mechanism for validation of the project. In this section, the teacher should provide students with a way to validate the project independently. That provides information in malfunction of the project.

A validation tool that runs the project with a series of tests designed is provided. This tool checks the correct operation of the project both outputs, outputs errors and memory leaks.

III. PROJECT IMPLEMENTATION

Once the project has been designed by the teacher, the teacher must make known this project to students.

In this section a number of general recommendations are given on how to deploy the project to help as a guide to their development. It also lists a series of steps the student should take to undertake the development and recommendations for code comments.

A. General recommendations

The work should be carried out in an incremental manner through successive refinement. The implementation of this project represents a different challenge because, for the first time, the development of a complex program is proposed. It must be designed and implemented individually; therefore, the solution is unique for each student.

The solution should be focused so as to follow these basic and interrelated principles:

- Simplicity: keep short, manageable programs.
- Clarity: it guarantees that are easy to understand for people.
- Generality: working well in a wide range of situations.

The following are general recommendations for addressing the problem solution:

- Review concepts already acquired of programming (control structures, files, data types, compilation and debugging tools, etc.).
- Start with the most basic.
- Follow a consistent style for indentation of the code and for naming variables and functions.
- Comment and document the code properly.
- Always choose the simplest solution.
- Be organized and structured.
- The better the code and documentation, more easily possible errors are detected and corrected.

B. Steps to develop the program that the student must perform

Following are a number of steps the student must follow to develop the project:

Step 1. Understanding the problem to solve. Understand the system to be implemented.

- Comprehensive reading of the statement and the problem proposed.
- Identification of the context and the different basic elements involved in the problem.

Step 2. Problem analysis and decomposition into smaller parts (Divide and Conquer).

Analysis and selection of the components needed to implement the solution:

- Data to be handled by the program.
- Tasks to be performed by the program.

Step 3. Design of system components.

In the next step, each of the features of the program has to be developed. To do this:

- In this step student makes a proposal structures data that the program uses. The choice of these structures determine the remaining implementation.
- Perform the tasks in functions. Each function, determines a name, inputs and outputs.

- Documentation of the design. Student describe in this step it implementation.

Step 4. Search for alternative solutions.

Evaluate the alternatives in terms of simplicity, clarity and optimization of resources (memory, runtime).

Step 5. Coding different parts.

- Define the data structures in the programming language.
- Split code in files, organizing it coherently based on logic and functional grouping.
- Coding and independent verification of each part (function). For each part to implement, perform the following steps:
- Coding.
- Documentation of the code.
- Evaluation of the complexity of the code.
- Test, independently, the correct operation of this part. Check borderline cases.
- Integration with the rest of the code.
- Test with the rest of the code.
- Reasoning of whether the implementation of the code can be improved.

Step 6. Full System Test.

Student should develop the tests needed to prove the satisfactory performance of the program implemented, including the limiting cases.

Step 7. Project Report.

Production of the report of the program, according to the rules.

C. Recommendations for code comments

The purpose of comments is to help the reader of a program. The best comments aid the understanding of a program with a brief indication of the important details, or offering a broader perspective on the code. The code comments should be written following these recommendations:

- Do not repeat what is already obvious.
- Discuss the functions and data types.
- Do not comment bad code. Rewrite it!
- Do not contradict the code.

Comments are useful to understand parts of the program. The better you do the code, less comments are necessary.

IV. CONCLUSION

In this paper we are shown the PBL learning methodology applied to the subject of Programming in Engineering. Shown are two views, teacher and student. First, the point of view of the teacher considers the design of the project, and secondly student that considers its implementation.

ACKNOWLEDGMENT

The authors would like to thank to all teachers of the both courses (FPI and FPII) Rafael Bachiller Soler, Isabel Román Martínez, Antonio José Estepa Alonso, Rafael Estepa Alonso, José Manuel Fornés Rumbao, José Ángel Gómez Argudo, Fernando Cárdenas Fernández, Pablo Nebrera Herrera, Godofredo Fernández Requena, Juan Manuel Vozmediano Torres, Antonio Luis Delgado González, and Germán Madinabeitia Luque. These results would not have been possible without their dedication.

REFERENCES

- [1] Allen, D. E., Duch, B. J., & Groh, S. E. (1996). The power of problem-based learning in teaching introductory science course. In L. Wilkerson & W. H. Gijsselaers (Eds.), *Bringing problem-based learning into higher education: Theory and practice*. San Francisco, CA: Jossey-Bass.
 - [2] Barak, M., & Dori, Y. J. (2005). Enhancing undergraduate students' chemistry understanding through project-based learning in an IT environment. *Science Education*, 89(1), 117–139.
 - [3] Barrows, H. S. (1996). Problem-based learning in medicine and beyond: A brief overview. *New Direction for Teaching and Learning*, 68, 3–12.
 - [4] Barnard, A., Nash, R., & O'Brien, M. (2005). Information literacy: Developing lifelong skills through nursing education. *Journal of Nursing Education*, 44(11), 505–510.
 - [5] Cunningham, W. G., & Cordeiro, P. A. (2003). *Educational leadership: A problem-based approach*. Boston, MA: Pearson Education.
 - [6] Fernández, F.J.; Sierra, A.J.; Ariza, T.; Madinabeitia, G.; Vozmediano J.M.; Virtualization Environment in Telematics Labs, Global Engineering Education Conference (EDUCON), 2013.
 - [7] Garland, N. J. (1995). Peer group support in economics: Innovations in problem-based learning. In W. Gijsselaers, D. Tempelaar, P. Keizer, E. Bernard, & H. Kasper (Eds.), *Educational innovation in economics and business administration: The case of problem-based learning* (pp. 331–337). Dordrecht, The Netherlands: Kluwer.
 - [8] Hung, W., Jonassen, D. H., & Liu, R. (2008). Problem-based learning. In J. M. Spector, M. D. Merrill, J. van Merriënboer, & M. Driscoll (Eds.), *Handbook of research on educational communications and technology* (3rd ed., Vol. 1, pp. 485–506). New York: Erlbaum.
 - [9] Oberlander, J., & Talbert-Johnson, C. (2004). Using technology to support problem-based learning. *Action in Teacher Education*, 25(4), 48–57.
 - [10] Ostwald, M. J., Chen, S. E., Varnam, B., & McGeorge, W. D. (1992). The application of problem based learning to distance education (ERIC Document Reproduction Service No. ED 359 398).
 - [11] Osgood, M. P., Mitchell, S. M., & Anderson, W. L. (2005). Teachers as learners in a cooperative learning biochemistry class. *Biochemistry and Molecular Biology Education*, 33(6), 394–398.
 - [12] Pletinckx, J., & Segers, M. (2001). Programme evaluation as an instrument for quality assurance in a student-oriented educational system. *Studies in Educational Evaluation*, 27, 355–372.
 - [13] Reynolds, F. (1997). Studying psychology at degree level: Would problem-based learning enhance students' experiences? *Studies in Higher Education*, 22(3), 263–275.
 - [14] Seltzer, S., Hilbert, S., Maceli, J., Robinson, E., & Schwartz, D. (1996). An active approach to calculus. In L. Wilkerson & W. H. Gijsselaers (Eds.), *Bringing Problem-based learning into higher education: Theory and practice* (pp. 83–90). San Francisco, CA: Jossey-Bass.
 - [15] Sierra, A.J.; Ariza, T.; Fernandez, F.J., Establishment EHEA for telecommunication technologies engineering degree Global Engineering Education Conference (EDUCON), 2012 IEEE Digital Object Identifier: 10.1109/EDUCON.2012.6201158 Publication Year: 2012, Page(s): 1–6.
 - [16] Sierra, A.J.; Ariza, T.; Fernandez, F.J.; Madinabeitia, G., TVSP: A Tool for Validation Software Projects in programming labs Global Engineering Education Conference (EDUCON), 2012 IEEE Digital Object Identifier: 10.1109/EDUCON.2012.6201196 Publication Year: 2012, Page(s): 1 – 6.
 - [17] Sierra, A.J.; Ariza, T.; Fernandez, F.J.; Madinabeitia, G., Tool for Validation Software Projects in Programming Labs, *International Journal of Engineering Pedagogy (iJEP)*, Vol 2, No 2 (2012).
 - [18] Smith, D., & Hoersch, A. L. (1995). Problem-based learning in the undergraduate geology classroom. *Journal of Geological Education*, 43, 149–152.
- A.J. Sierra** is with the Department of Telematic Engineering at the School of Engineering of the University of Sevilla, Camino de los Descubrimientos S/N, 41092 Seville, Spain (e-mail: antonio@trajano.us.es).
- T. Ariza** is with the Department of Telematic Engineering at the School of Engineering of the University of Sevilla, Camino de los Descubrimientos S/N, 41092 Seville, Spain (e-mail: matere@trajano.us.es).
- F.J. Fernández** is with the Department of Telematic Engineering at the School of Engineering of the University of Sevilla, Camino de los Descubrimientos S/N, 41092 Seville, Spain (e-mail: fjfj@trajano.us.es).