PLOP: A Pattern for Learning Objects for Programming

Luis A. Álvarez-González, Valeria Henríquez N., Erick Araya and Fabiola Cárdenas
Grupo de Investigación en Tecnología de Aprendizaje
Universidad Austral de Chile, Valdivia, Chile
lalvarez@inf.uach.cl, valeria.a.henriquez@gmail.com, earaya@uach.cl, fabiolacardenas@uach.cl

***Abstract*—This paper shows a Pattern of Interaction to enhance the construction process of learning objects for introductory programming languages. All the phases of the design of the pattern are showed. In the phase of educational design, some elements of software engineering are included. In the phase of multimedia design, the information areas are showed and also the sub-areas for each information area. In the implementation phase, the class diagrams of the pattern are showed. Finally, the evaluation and conclusions are presented.**

Keywords-components; learning objects; interacction patterns; programming languages.

I. INTRODUCTION

At the Universidad Austral de Chile, located in Valdivia, south of Chile, it is important to incorporate new teaching methods to enhance learning achievement of their students, especially in introductory programming courses where it has been detected a low level of approval [1], 69% the first, 53% second, the third 55% and 54% in the fourth semester (courses Info030, Info033, Info043 and Info053 in Figure 1). The data presented clearly demonstrates the need to develop educational resources to support teaching, especially for struggling students during their first steps in programming, as well as support teachers in the production process of educational material.

Three projects focused in building learning objects (hereafter LOs) for programming languages were found as the most important:

- The CodeWitz Project which aims to plan, produce and evaluate illustrations, animations and visual support for programming students and lecturers [4]. In the CodeWitz network are participating 29 universities in 21 countries, only the universities of the network can use the CodeWitz LOs.
- The Project "A Programming System Education Based on Program Animation", Gakugei University, Tokyo, Japan, which aims to support students with problems and help them understand the performance of different algorithms, with particular emphasis on implementation and changes occurring in the data structures in memory [6].
- Learning Objects for Introductory Programming Project, developed at London Metropolitan University by the staff in the Learning Technology Research Institute (LTRI) and the Department for Computing, Communications Technology & Mathematics, and at Bolton Institute [1].

However, the LOs of these three projects are not available to any user, neither have a good interaction, moreover they do not have areas to define problems, diagrams or metadata, they are written in English and to build a new one the user must modify an old one. In other words, there is not a tool to build LO.

Then, how the lecturers can be converted into producers of LOs? To answer this question we developed a pattern of interaction and a methodology that allows the creation of new LOs following a repeatable process that can be used by any lecturers. With these LOs, it is intended that students can follow the execution of program at a code level, step by step in an interactive fashion to be able to learn for themselves by promoting metacognition, autonomy and respecting the learning pace, regardless of their location (available on Internet).

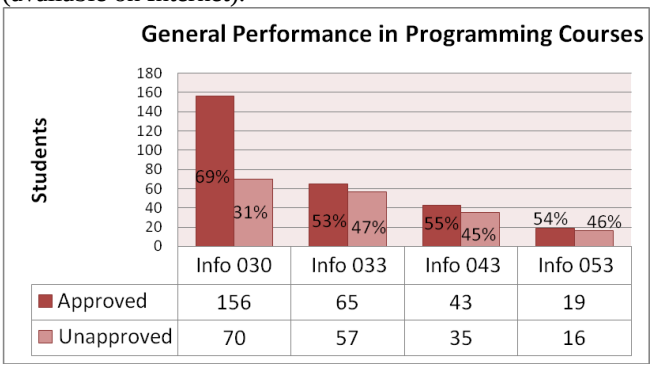

Figure 1. Approval rate in introductory programming courses.

II. A PATTERN OF INTERACTION

A Patterns of Interaction is understood as an effective solution to a recurring problem, because it promotes the reuse of good design, shortens development time and captures the experience of expert designers and programmers in the development of usable interfaces. They

[1] Information given by the School of Informatics Engineering of Universidad Austral de Chile, in the period 2006-2008.

 ISBN: 978-1-61208-111-3

condense the experience into a series of guidelines or recommendations, which can be used by novice developers to the purpose of acquire the ability to design users interfaces [3]. To describe the pattern of interaction, the definition of Welie [8] is used and it is shown in Table 1.

TABLE 1. DESCRIPTION OF THE PATTERNS OF INTERACTION.

Item	Description
Name	Represent the execution of an algorithm
Author	Valeria Henríquez-Norambuena
Problem	How to simulate the execution of an algorithm?
Usability Principle	Reduce cognitive charge
Context	This pattern can be used in all educative situations, where you want to understand a running algorithm
Forces	Representation like debug is easy to understand for students
Solutions	Show diagram, memory, I/O and help areas, which can represent the algorithm execution.
Consequences	Develop LOs using this pattern, allows to create a mental image about what happens into the computer when the code is run.

The design of the pattern is based on the methodology of the University of Guadalajara [2].The stages of this proposal are: Educational Design, Interaction Design, Functional Design, Multimedia Design, Implementation and Phase Labeling and Packaging.

A. *Educational Design*

The educational design is based on the methodology developed by the LO Group of the Universidad Autónoma de Aguascalientes [7]. Lecturers are the creators and developers of LOs, assistants and students can play the role of developers, both actors with algorithmic knowledge and training. Like any methodology that uses patterns, you need a bank of patterns for your application. It defines the stages and in each of the activities, artifacts and actors involved. The process includes four phases: Analysis, Development, Testing and Implementation. Figure 2 shows in detail this methodology.

1) Analysis Phase. At this stage, it is important to identify the competences we want to develop. Based on this, the author of the content (lecturer), describes in the analysis document, what are the general requirements of the LO and obtains the necessary materials. It is essential that the requirements document is sufficiently clear so that any other parties can continue the process of creating the LO.

a) Activities: To analyze the problem and to obtain the necessary materials.

b) Artifacts: Requirements document.

c) Actors: Author (lecturers).

2) Development Phase. The developer selects a pattern that meets the needs identified in the Requirements Specification artifact, generated in the previous phase. After selecting an appropriate standard, we make use of it. This activity is carried out as joint work between the author and coach. The pattern is "filled" with the material made by the author, but contextualisation is provided by the developer. The collaborative work between the actors (author and developer) through regular meetings is likely to decrease the time of the evaluation stage.

a) Activities: To select and to use the pattern.

b) Artifacts: LO and Metadata.

c) Actors: Author (lecturer), developer (programmer).

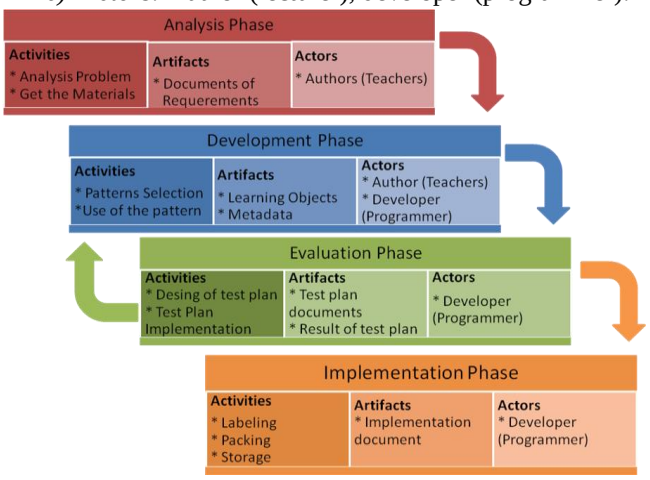

Figure 2. Educational Design Diagram.

3) Evaluation Phase. This phase assesses the proper functioning of the LO, i.e. the relevance of content and design. For this phase a plan which specifies the test cases and the results expected is produced. On error, the LO will return to the stage of development to correct the problems identified.

a) Activities: To design and to implement a test plan.

b) Artifacts: Test plan document, results of the test plan report.

c) Actors: Evaluator (lecturer).

4) Implantation Phase. After the evaluation phase, the LO is labeled, packaged and stored. At this stage it is important to define the standard to be achieved in the packaging of LO. Then stored in a repository of LO, from which it can be downloaded and/or performed for the interaction and subsequent use.

a) Activities: To label, to package and to store the LO.

b) Artifacts: A document specifying details of the implementation of the LO.

c) Actors: Technical (programmer).

B. *Interaction Design*

For the construction of the pattern of LOs it was decided that the areas of pattern information are as follows:

1) Diagram Area: Contains the description of the problem and the graphical solution using a diagram.

2) Code Area: Contains the solution in a programming language and simulates the execution of the code. This area contains the following subareas:

 ISBN: 978-1-61208-111-3

a) Implementation Area: Solution in a programming language.

b) Execution Area: Simulation of the potential window displayed by the user.

c) Memory Area: Simulation of the process that occurs in the computer's memory.

d) Explanation Area: Explanation of each code instruction.

e) Control Area: Container buttons.

3) Metadata Area: Contains information related to the LO.

C. Overview of Pattern Operation

- The program will be run step by step through the code area, being the student who decides when to execute the following line of code.
- The program can run automatically and be interrupted if the student decides so.
- The rewind of the code execution is allowed.
- The line of code will be clearly identified.
- Each instruction is synchronized with the support area that will display an explanation for the line running.
- What it is displayed in the memory and execution areas will depend on the line of code
- A LO developed using the interaction pattern must cover any basic problem of programming, in any programming language.

D. Functional Design

To use the pattern, the Adobe Flash CS4 is required. The implementation requires a standard Web browser with Adobe Flash Player, which is available in most browsers. Because the files have `.swf` extension, a LO can be executed in any video player.

E. Multimedia Design

For multimedia design of the different components of the pattern, a model is created. To locate the components, is considered the usual reasoning to read, that is, from left to right and from top to bottom. The model contains two sections of display: the first one, navigation section, which contains three areas ordered by importance to the student: diagram area, code area and metadata area. The second one, located under the navigation section, shows the information for each area (Diagram, Code and Metadata). Figure 3 shows the reasoning in the design.

Figure 4 shows the Diagram Area, which consists of two parts: the left section, an area which describes the problem to be solved and the right section, an area that contains an image with a flow diagram to solve the problem. The order shown allows the student to read the description of the problem and then imagine a solution in a diagram on the right side.

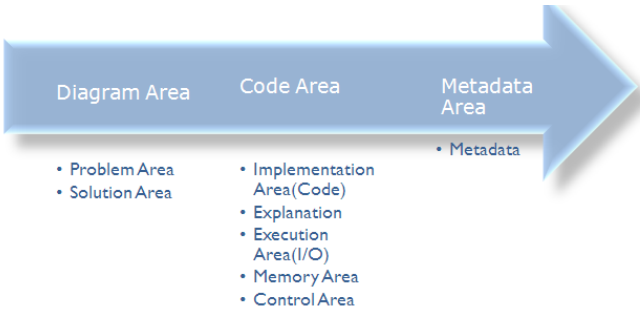

Figure 3. Sections Navigations of the pattern.

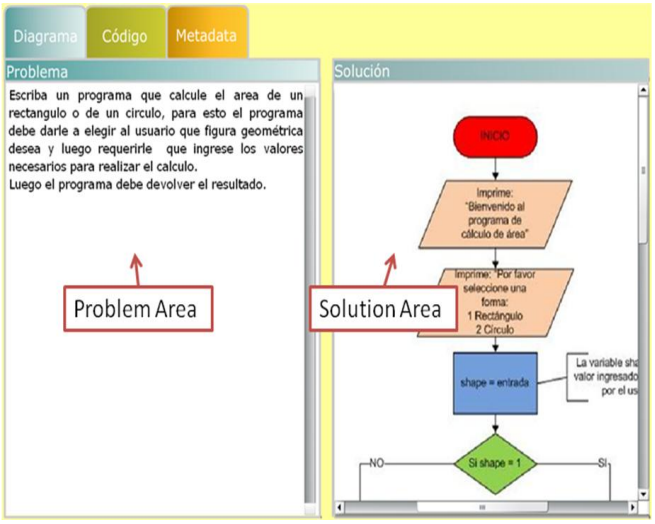

Figure 4. Diagram area.

Figure 5 shows the Code Area that contains other five areas described in order of importance:

- Implementation Area, which contains text that represents the solution in a programming language.
- Explanation Area contains text synchronized with the Implementation Area that displays information explaining the statement being executed in the code.
- Execution Area. It may contain text or images that simulate the inputs and outputs of the algorithm. This area is the third in importance, because it gives the student an idea of what should happen on the screen of the computer when the code is run.
- Memory Area. Contains animations or images that simulate the processes occurring in the computer memory when the code is run. These changes are not obvious to the student, therefore, lies in fourth place of importance.
- Control Area. Contains buttons that control the implementation of LO.

Figure 6 shows the area of metadata. This area contains a table with information related to the LO. For the metadata was considered a subset of the fields suggested by the standard committees of the IEEE Learning Technologies [5].

 ISBN: 978-1-61208-111-3

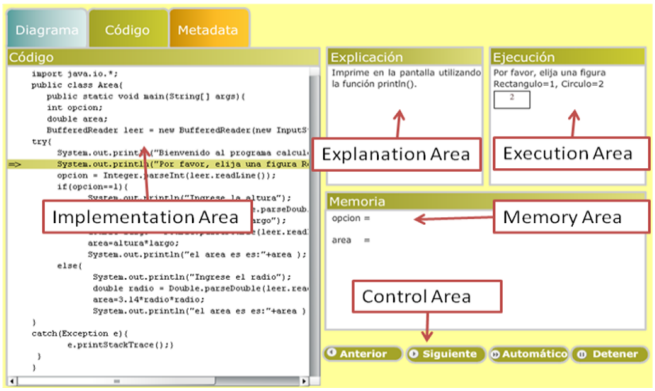

Figure 5. Code area.

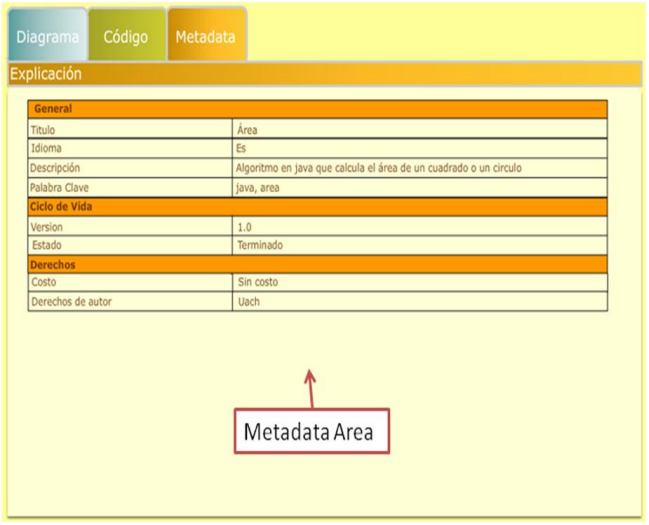

Figure 6. Metadata area.

F. *Implementation*

After the previous stages a LO is built, using Adobe Flash CS4 tools and the ActionScript 3 programing language. For the correct operation of the pattern three files are necessaries (see Figure 7): one with a `.fla` extension, which contains the pattern design, and two files with `.as` extension: **`Patron.as`** that handles the functionality of the standard and **`Mensaje.as`** controls the display of error messages (for example data entry is required and the user does not enter any value).

G. *Labelling and Packaging*

For the development of the pattern of interaction the labeling and packaging process was modified to the creation of a file with `.rar` extension, that contains the pattern and also four LO samples (`HelloWorld`, `HolaNombre`, `Addition` and `Selection`). The sample LOs were built during a validation workshop.

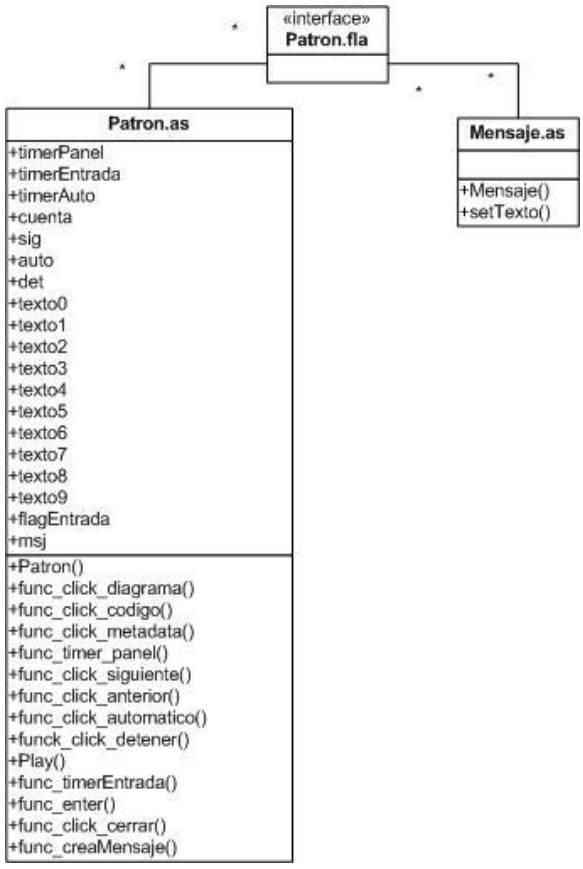

Figure 7. Pattern Class Diagram.

III. VALIDATION OF THE METHODOLOGY AND PATTERN OF INTERACTION

The validation of the proposed methodology and the pattern of interaction were done through the completion of two workshops and involved a group of six lecturers and four assistant students. Most lecturers give classes in programming at the same university.

A. *Validation Workshop*

The validation workshop was practical and considered three stages.

- Exposure of the general aspects, of the motivations and descriptions of the interaction pattern, including the educational design.
- Practical activities. Participant lecturers took the role of each actor, i.e., educational designer, technical developer and students following a LO. All these activities were conducted in order to gain a better understanding of the methodology (educational design) and the pattern.
- Instrument validation. Implementation of an online survey to participants of the workshop was applied. The survey form contains two sections:

The first seeks to validate the three criteria to be considered when using the pattern of interaction, that is, areas of information, multimedia content and the interaction of LO with the student.
The second refers to the pedagogical design, which seeks to validate the possible impact of using it in the development of LO.
This was aimed at validating this project as a solution to the identified needs.

B. Validation Results

Considering nine responses of ten participants in the workshops and a score from 1 to 5, according to Likert's scale (http://es.wikipedia.org/wiki/Escalas_Likert), being 1 strongly disagree and 5 strongly agree. The survey results of the interaction pattern can be seen in Table 2., the average assessment was 4.2 points. It should be noted that the question N° 2 related to sub-areas of information, look better evaluated with 4.7 points.

TABLE 2. SURVEY RESULTS OF VALIDATION PATTERN

Nº	Questions	Score
1	The three areas of information (Diagram, Code and Metadata) seem sufficient for learning a basic algorithmic problem?	4,4
2	The subareas (Problem, Solution Diagram, Solution, Explanation, Implementation, Memory and Control) seem sufficient for learning a basic algorithmic problem?	4,7
3	Do you think the use of the PLOP interaction pattern for the creation of learning objects may help improve the performance of struggling students in introductory programming courses?	4,0
4	The interaction with the areas and subareas are natural for you?	4,0
5	Do you think that the images, text and buttons used are appropriate?	4,4
6	Do you think the use of the pattern of interaction could be enhance in terms of productivity, the creation of learning objects for introductory programming?	3,9
	Average	**4,2**

The survey results for the educational design are shown in Table 3; the average evaluation was 4.05 points.

TABLE 3. SURVEY RESULTS OF EDUCATIONAL DESIGN

Nº	Questions	Score
1	Do you think that the phases of the methodology are appropriate to build Learning Objects?	4,0
2	Do you think that the use of this methodology could improve productivity in developing Learning Objects?	4,1
	Average	**4,05**

Note that this methodology cannot be compared with other similar projects [1][4][6], since there are no known existing projects on patterns applied to learning objects for programming teaching. In the Codewitz Project [4] most learning objects are built using the Macromedia Director (multimedia application authoring platform). To build another one, usually a previous learning object is used. For the other projects [1][6] there is not available information about the building. In other words, none of the previous projects use patterns.

IV. CONCLUSIONS

Considering that only three projects were found with similar objectives, a pattern of interaction in Spanish is built. This pattern considers significant improvements in the areas of information (content distribution), design (colors, buttons, etc.) and metadata for LO. The pattern is enhanced and adapted from the pattern developed by Delgado, Morales, Gonzalez and Chan. The enhancement is because the methodology proposed by Osorio, Muñoz and Alvarez was improved considering software engineering aspects and is included in the pattern. The adaptation is because the pattern and methodology is developed for programming languages learning objects.

The proposed interaction pattern corresponds to a tool for building LOs to support computer programming, so that lecturers can improve approval ratings. Given this context, lecturers are relatively agreeing (Average score 4,2 in Table 2.) that the pattern can serve to build LOs which to improve their students' learning. However, when asked whether the subareas are sufficient, the answer is very close to maximum (4.7 on a scale of 1-5). They also give a good score the areas of information, as well as images, text and buttons. But, do not completely agree (3.9) that the proposed pattern can help to improve the productivity of LOs. Based on this analysis we can conclude that new support tools for the construction of LOs are required. These news support tools should be easier to use, such as frameworks or templates, but keeping the design, and sub-areas of information.

Moreover, the teaching methodology (Educational Design) was not sufficiently well evaluated (4,05 in Table 3.), according to the authors, this is due to the absence of previous patterns which affects the phase II of the teaching methodology (Development).

Finally, the validation workshop concluded that to develop a low complexity LO, a lecturer required about three hours including pedagogical design.

V. ACKNOWLEDGEMENTS

The authors wish to thank the Research and Development Office of the Universidad Austral de Chile through project No. S-2007-15 entitled Learning Designs and Classroom Management, project DID S-2007-02 entitled Algebra Learning Units using ICT. Additionally wishes to acknowledge the valuable comments of all members of the Research Group on Learning Technologies (www.gita.cl).

REFERENCES

[1] Bradley, C. and Boyle, T. (2004). "Student evaluation of the use of learning objects in introductory programming". In L. Cantoni & C. McLoughlin (eds.), proceedings of ED-MEDIA 2004, World Conference on Educational Multimedia, Hypermedia & Telecommunications, June 21-26, Lugano, Switzerland, AACE, pp. 999-1006, ISBN 1-880094-53-3.

[2] Delgado J., Morales R., González S., and Chan M., (2007), "Desarrollo de Objetos de Aprendizaje basado en patrones". Sistema de Universidad Virtual, Universidad de Guadalajara. México. Retrieved

from http://ihm.ccadet.unam.mx/virtualeduca2007/pdf/228-JDV.pdf [last access:10.04.2010]

[3] Hernández, M., Alvarez. G., and Muñoz. J. "Patrones de Interacción para el Diseño de Interfaces WEB usables". Retrieved from http://hciinterfazbuscador.iespana.es/PatronesInteraccionDiseno.pdf [last access:10.04.2010].

[4] Kujansuu, E. and Tapio, T. "Codewitz – An International Project for Better Programming Skills". In L. Cantoni & C. McLoughlin (Eds.), Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2004 (pp. 2237-2239). Chesapeake, VA: AACE. Retrieved from http://www.editlib.org/p/12334 [last access:10.04.2010].

[5] Learning Technology Standards Committee of the IEEE. (2002): "Draft Standard for Learning Object Metadata. IEEE 1484".12.1-2002. Online version. Retrieved from : http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf [last access:10.04.2010]

[6]. Miyadera, Y., Kurasawa, K., Nakamura, S., Yonezawa, N., and Yokoyama, S. "A Real-time Monitoring System for Programming Education using a Generator of Program Animation Systems" JOURNAL OF COMPUTERS, VOL. 2, NO. 3, MAY 2007. pp. 12-20

[7] Osorio, B., Muñoz, J., and Álvarez, J., (2007). "Metodología para el desarrollo de objetos de aprendizaje usando patrones", 2da. Conferencia Latinoamericana de Objetos de Aprendizaje, Santiago de Chile (2007). Retrieved from : http://www.laclo.espol.edu.ec/laclo2007/index.php?option=com_content&task=view&id=26&Itemid=50&lang=en [last access:10.04.2010]

[8] Welie, M., (2000), "Patterns as Tools for User Interface Design", Vrije Universiteit, Department of Computer Science, Amsterdam. The Netherlands, Retrieved from : http://www.welie.com/papers/TWG2000.pdf [last access:10.04.2010].

 ISBN: 978-1-61208-111-3