REVISTA COLOMBIANA DE COMPUTACIÓN
Volumen 13, número 2
Págs. 7 - 18

Virtual Learning Environment to Support Object Oriented Programming Learning

Adriana M. Angarita*‡, César D. Guerrero*, Graciela Perera**

Fecha de recibido: 28/06/2012 Fecha de Aprobación: 12/10/2012

Abstract

Object-oriented programming is one of the core programming paradigms taught in introductory programming courses in engineering. It demands learning strategies that support the understanding of its concept and the implementation of these in computer programs. We present a Virtual Learning Environment (VLE) used to overcome the difficulties identified by students and professors. These difficulties were collected during their participation in an Object Oriented Programming (OOP) class. VLE allowed students to self-evaluate the concepts and skills gained in class using a Blended Learning strategy. This includes performing on-line activities that can improve or solve their theoretical or practical gaps in a particular OOP subject. VLE is implemented in DOKEOS and evaluated by first-year engineering students at *Universidad Autónoma de Bucaramanga*. Results show that by implementing additional learning strategies such as student-centered learning, an effective outcome is achieved when teaching OOP.

Keywords: *E-Learning, Object oriented programming, Dokeos.*

*Universidad Autónoma de Bucaramanga, Colombia. aangarit@gmail.com, cguerrer@unab.edu.co
** Northeastern Illinois University, USA. G-Perera@neiu.edu

1. Introduction

In recent years, there has been a significant reduction in the student enrollment to engineering programs. There are several known reasons to explain this trend. This includes the lack of mathematical background, the budget reduction in science and mathematics education by some governments, and the perception by young people who think there may not be jobs in this field. This trend is opposite to the number of jobs that employers can offer. For example, in the immediate future of 2020 there will be 20 million jobs .

Lack of mathematical background is not the only reason for students to not consider or leave engineering programs. Some students experience learning difficulties in computer programming courses which are an essential part in the engineer education. Furthermore, introductory courses in computer programming require pedagogical and technological strategies to attract and not reject students. These courses should be helping student acquire foundations of good programming; "The correct programming does not fall from the sky" [5]. Object-oriented programming is one of the most influential programming paradigms in recent years [7] and demands learning strategies that help students understand the concepts and implement them in computer programs.

This paper proposes an e-learning strategy that can help students taking an introductory course in OOP. The strategy is developed as a supporting technology for the physical learning environment and for that reason can be considered as a blended learning environment strategy.

The rest of the paper is organized as follow. In Section II it is drawn the state of the art making emphasis on object oriented programming learning strategies. Section III shows a preliminary study performed to determine the main learning difficulties as depicted by students and professors in OOP courses in four local universities. Section IV describes the Virtual Learning Environment proposed in this paper as a way to enhance the student performance in an OOP course. In Section VI the approach proposed is evaluated in a OOP course at *Universidad Autonoma de Bucaramanga*. Finally, in Section VII conclusions and future work are depicted.

2. State of the art

Computer programming education is taking advantage of e-learning implementations from different authors and methodologies. Some of the main approaches in the field are the following: Misoko [3] aims to implement Web-based teaching by personalizing learning. This author provides a way to improve programming learning by using the method of "watching others". With this purpose, tools were created where students write a piece of code and receive feedback from the teacher using the same tool. This allows students to create code, identify errors, and receive guidance and support from their teachers. This study shows the convenience of using computer programming e-learning tools.

Another work developed at Queensland University [9] refers to an Environment for Programming Learning (EPL) which provides a web-based interactive environment to teach computer programming. The University has made it easier for students to work collaboratively. This benefits both teachers and students as it encourages feedback, allow students to practice and progress at their own pace. EPL provides the possibility of coaching at anytime and anywhere by the easy integration of notes, tutorials and practical exercises. This study shows the strength of the web-based education in solving several problems that often occur in the classroom when teaching programming using Java.

Universities in Taiwan have joined efforts to formalize a study presented by Yuin Wu [4] where a Web-based solution called "DRW" offers five programming activities with multiple difficulty levels based on the Bloom's taxonomy. DRW provides online code, execution and annotation tools to conduct training and assignments via Web-based programming. It addresses not only a tool but also a complete methodology through activities designed for the Web.

In the Department of Computer Science at the University of Warwick [8] a study describes collaborative e-learning with a co-evaluation strategy performed by peers (students). It has been found that constructive criticism against the work of others encourages students to improve their efforts towards correcting their mistakes by watching others doing their job. Also, it presents some features to be improved such as the possibility of affecting student's subjectivism in assessing the work of others.

According to Bruce [2], students can learn to program in five different ways: 1) Following or students learn by experience; 2) Building Code, that is students learn programming by coding; 3) Understanding and integrating, where the act of programming is experienced through the understanding and integration of concepts; 4)Problem Solving, which is experienced through learning what it takes to solve a problem; and 5) Participating, which is learning by experience when being a programmer in the field.

One of the different OOP strategies and methodologies that have been implemented is called "Object-first" which introduces students to the concepts, classes and instances before procedural elements [13]. A simplistic view of a holistic development leads to include ideas and concepts gradually instead of having an integrated perspective from the outset [1].

It is important to note that there are conditions for effective use of new methodologies. Adopters need to have an advanced level of Information Communication Technology (ICT) use and design, as well as teaching skills and project management. It is also desired a good learner to be motivated, expertise by the teacher of what to teach, teaching tools and proximity to the reality as a means of connecting with experiential learning [10].

A very interesting study developed in Colombia analyzed and designed a First Course in Computer Programming called CUPI2 project [11][12] This project consisted of an initial survey that collected all the experiences obtained in programming courses at a University in Colombia. Findings address the implementation of different strategies and the study of possible reasons that explain why some groups may be more successful than in others.

When looking at Object Oriented Programming (OOP), there are also several documented difficulties. These include issues on decomposing a problem, understanding the mechanics of programming, passing parameters, returning values of methods and variable declarations, abstract programming techniques used in different situations, and finally understanding and using object-oriented programming concepts such as inheritance, abstraction, classes and interfaces [6]. Problems in teaching OOP are due to bad choice of languages and wrong teaching aids [7]. According to Kölling, a suitable environment must support seven key areas: ease of use, integrated tools, support objects, support code reuse, support learning, support teams, and availability [7].

3. Problem Identification

In order to identify the problem of teaching and learning OOP an online survey was conducted. The survey gathered anonymous feedback from engineering students who have completed their first semester and first contact with OOP. The survey was implemented using *LimeSurvey* an applied to students from four Colombian universities: *Universidad Autónoma de Bucaramanga*, *Universidad de Santander*, *Universidad Pontificia Bolivariana*, and *Universidad*

Industrial de Santander. The study was focused on the following five topics: 1) prior experience in programming courses, 2) work or personal programming experiences, 3) degree of complexity of major programming issues, 4) importance of the strategies employed when learning programming, and 5) evaluation of programming topics foundations. Tables 1 and 2 show the results of the survey.

Topic	Very Low	Medium	Very High
Algorithm Concept	28.57%	30.95%	11.90%
Build algorithms or flow charts	21.43%	33.33%	14.29%
Relational operators, arithmetic and logical	14.29%	35.71%	9.52%
Selection Structure Single, Double and Nesting	11.90%	26.19%	1667%
Repeat structures	16.67%	35.71%	19.05%
Numbering systems and conversion	7.14%	30.95%	23.81%
Introduction to Java	9.52%	35.71%	28.57%
Builders and Definition	11.90%	42.86%	28.57%
Concepts and application of methods	7.14%	42.86%	28.57%
Basic operations on arrays	9.52%	35.71%	28.57%
Object Oriented	9.52%	26.19%	33.33%
Arrange step methods	7.14%	28.57%	35.71%

Table 1. Degree of complexity for each item in a Fundamentals of Programming course.

The survey was open to students for about 20 days to complete 50 samples.

Topic	Very Low	Medium	Very High
Exercises on board	4.76%	30.95%	42.86%
Practice Exams	0.00%	26.19%	40.48%
Online quizzes (online)	9.52%	30.95%	21.43%
Online Tutorials (online)	0.00%	9.52%	30.95%
Programming in teams of 2 student s	14.29%	16.67%	28.57%
Tutoring a student of by a students from higher semesters	9.52%	26.19%	28.57%
Teacher's office hours	2.38%	9.52%	47.62%
Alice	48.78%	14.63%	12.20%
Programming Videos	4.76%	26.19%	28.57%
Support through monitors	2.38%	23.81%	38.10%
Practical work with classroom projects throughout the course	0.00%	19.05%	61.90%
Analysis and study of software provided	0.00%	23.81%	52.38%

Table 2. Degree of importance of the strategy used to improve computer programming learning.

Since, most of the strategies can be implemented in a VLE, the results shown in Table II illustrate the feasibility of our approach and the belief that we can achieve positive results. Teachers from the universities mentioned above and others located outside the area where also interviewed. They shared their experience when teaching OOP. Seven of them explained the major concerns with respect to two specific subjects: the aspects that make teaching programming a very complex task and the teaching strategies used by them.

From students' surveys and teachers' interviews, several elements were collected as a baseline and as a starting point for the development of the e-learning strategy. Some of them are the following:

- Need for alternative learning strategies
- Need of project-based learning
- List of topics to be included in the strategy; these were the most difficult for the students.
- Student's initial background in logic, algorithms, analysis and abstraction techniques. Also, reading comprehension and mathematical concepts are key towards student success.
- It is very important to emphasize in programming courses the problem resolution methodology used by students and the design of the algorithms.

4. Virtual learning environment

In this section it is described the VLE approach used in this paper. One of the most important considerations in the production of a teaching tool is the clear identification of the end users. The VLE developed is expected to be used by students who are digital natives and find it very easy to use technological tools. They are engineering students who know the use of Web 2.0 tools. Specifically, the design of the tool considers the following assumptions about the students: they are autonomous and independent; they understand the main concepts of Object Oriented Programming; they have good listening and comprehension English skills; they are familiar with the use of Web 2.0 tools; and they are prompt to develop collaborative work with peers.

Based on the diagnostic depicted in Section III, the main characteristics of the VLE approach used in this work are the following:

1. The main subjects of Object Oriented Programming are treated as modules. This allows better assimilation of the content by viewing it sequentially and in order.

2. Use of a free online platform that allows students to review as many times as needed the topic of their choice using a communicative approach and collaborative work groups.
3. Begin with initial tests that enable students to observe and obtain a qualitative view of their understanding of the topic studied. Also, identify the issues to be improved.
4. Provide multimedia content covering the topics in a meaningful way and allowing students to repeat a topic as many times as necessary.
5. Propose practical exercises focused on the teaching needed to establish simple relationships between reality and layout of the solution
6. Divide the learning process in five stages having each stage a learning object, learning activities, and two tests to evaluate the learning progress. One test will be presented at the beginning for assessment and to establish the initial state of learning. The final test will allow to evaluate the effectiveness of the VLE when compared with the initial test.

The VLE proposed has five major aspects to effectively track the student's learning process and to offer more control for the teacher to address learning strategies and implement them. That also allows to be focused on the learning objectives for each topic and to take advantage of the reuse of course materials. These aspects are the following:

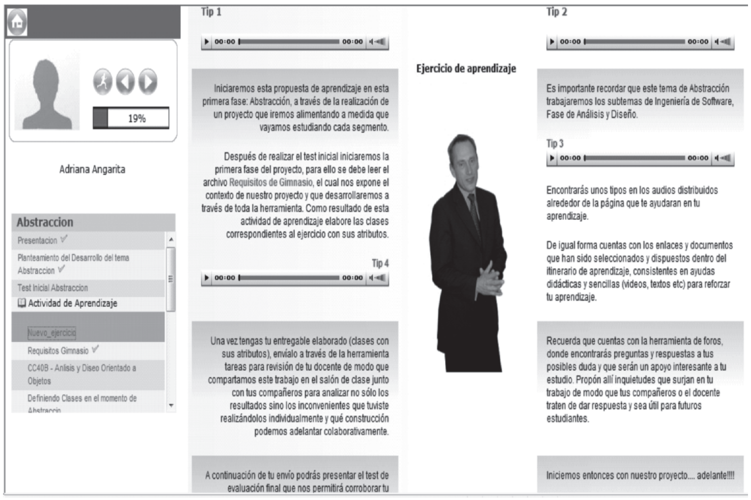
- **Abstraction**: Software Engineering, analysis and design phases.
- **Implementation**: Declaration of attributes, methods, hiding information, access to constructors, and encapsulation.
- **Class Design**: Inheritance, over-writing methods, access control to methods, overloading methods, and data type conversion.
- **Advanced**: class attributes, class methods, variables, static declarations, abstract classes and interfaces.
- **Java**: Development in the language (coding).

The proposed VLE has its main support in collaborative and cooperative work, based on autonomous and critical learning to make the student aware of his or her learning through a classroom project. This project was developed based on the response obtained in the surveys where students gave high importance to develop projects in the classroom as an effective way of learning.

The Learning Management System used to implement the five major aspects described before was DOKEOS. This platform allows to implement the VLE as a scheduled sequence of learning objects. The student uses the VLE with a close support by the professor in the physical class sessions and through the reports provided by the tool. This follow up is performed at different times to make sure that the student can go further after achieving determined goals and objectives for each module.

As mentioned before, the learning path begins in a first phase with the process of Abstraction. The student starts taking a short five-question test in the tool that can be submit as many times as required. The tool provides a clear and accurate feedback of the errors and the reasons why a given answers is not correct. This allows students to learn from their mistakes and find themselves a solution to the test presented.

Following the test, a learning activity is presented as the implementation of the first phase of a project related to the issue of abstraction. Here, the student needs to define the context where data will be implemented. To carry out this learning activity the student will have the collaborative tools that DOKEOS makes available: forums, Wikis, Blogs and chat. Figures 1 to 3 show a few screenshots of the VLE.



Iniciaremos esta propuesta de aprendizaje en esta primera fase: Abstracción, a través de la realización de un proyecto que iremos alimentando a medida que vayamos estudiando cada segmento.

Después de realizar el test inicial iniciaremos la primera fase del proyecto, para ello se debe leer el archivo Requisitos de Gimnasio, el cual nos expone el contexto de nuestro proyecto y que desarrollaremos a través de toda la herramienta. Como resultado de esta actividad de aprendizaje elabore las clases correspondientes al ejercicio con sus atributos.

Una vez tengas tu entregable elaborado (clases con sus atributos), envíalo a través de la herramienta tareas para revisión de tu docente de modo que compartamos este trabajo en el salón de clase junto con tus compañeros para analizar no sólo los resultados sino los inconvenientes que tuviste realizándolos individualmente y qué construcción podemos adelantar colaborativamente.

A continuación de tu envío podrás presentar el test de evaluación final que nos permitirá corroborar tu

Es importante recordar que este tema de Abstracción trabajaremos los subtemas de Ingeniería de Software, Fase de Análisis y Diseño.

Encontrarás unos tipos en los audios distribuidos alrededor de la página que te ayudaran en tu aprendizaje.

De igual forma cuentas con los enlaces y documentos que han sido seleccionados y dispuestos dentro del itinerario de aprendizaje, consistentes en ayudas didácticas y sencillas (videos, textos etc) para reforzar tu aprendizaje.

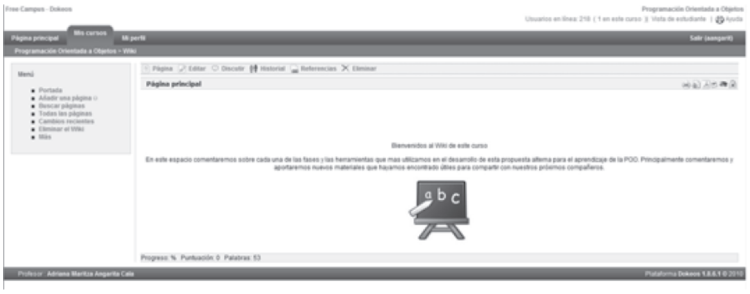
Recuerda que cuentas con la herramienta de foros, donde encontrarás preguntas y respuestas a tus posibles duda y que serán un apoyo interesante a tu estudio. Propón allí inquietudes que surjan en tu trabajo de modo que tus compañeros o el docente traten de dar respuesta y sea útil para futuros estudiantes.

Iniciemos entonces con nuestro proyecto.... adelante!!!!

Fig. 1. A VLE Lesson page.

Once the student completes all the exercises in the module, he or she presents a test similar to the initial 5 questions. With this final test, students can be aware of the knowledge gained through the tool. This provides initial statistics for validation and evaluates the effectiveness of the approach.

Similarly, five work moments corresponding to the five aspects considered by the VLE are synchronized with the instants in which students cover every studied aspect in the class.

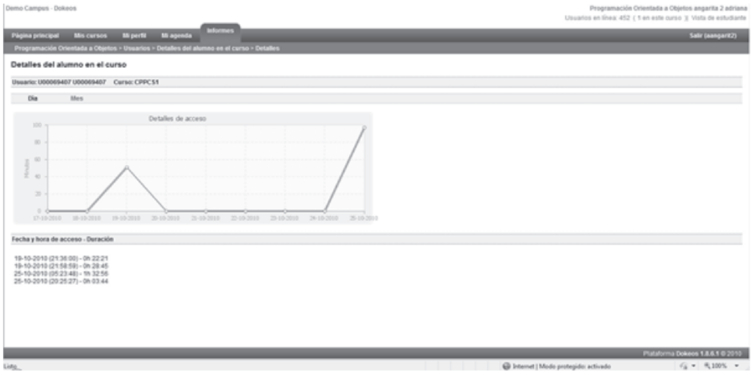


Bienvenidos al Wiki de este curso

En este espacio comentaremos sobre cada una de las fases y las herramientas que mas utilizamos en el desarrollo de esta propuesta alterna para el aprendizaje de la POO. Principalmente comentaremos y aportaremos nuevos materiales que hayamos encontrado útiles para compartir con nuestros próximos compañeros.

Progreso: % Puntuación: 0 Palabras: 53

Fig. 2. A VLE Wiki.


Demo Campus - Dokeos
Programación Orientada a Objetos angarita 2 adriana
Usuarios en línea: 452 (1 en este curso) Vista de estudiante
Página principal
Mis cursos
Mi perfil
Mi agenda
Informes
Salir (aangarit2)
Programación Orientada a Objetos > Usuarios > Detalles del alumno en el curso > Detalles
Día
Mes
Detalles de acceso
Minutos
100
80
60
40
20
0
17-10-2010
18-10-2010
19-10-2010
20-10-2010
21-10-2010
22-10-2010
23-10-2010
24-10-2010
25-10-2010
Plataforma Dokeos 1.8.6.1 © 2010
Listo
Internet | Modo protegido: activado
100%

Detalles del alumno en el curso

Usuario: U00069407 U00069407 Curso: CPPCS1

Fecha y hora de acceso - Duración

19-10-2010 (21:36:00) - 0h 22:21
19-10-2010 (21:58:59) - 0h 28:45
25-10-2010 (05:23:48) - 1h 32:56
25-10-2010 (20:25:27) - 0h 03:44

Fig. 3. Control Access Details for one student.

5. Students perception of the designed VLE

This section shows the main results of conducting an evaluation test of the VLE developed with a group of 30 engineering students. The group presented the initial test focused on the theoretical Abstraction subject. The results showed a very low performance in this subject. After that test, the student proceeded to perform the learning activity and send it through the tool; this activity was only made by 20% of the students (six of them).

Another learning activity was the presentation of a video from an expert of the discipline. This expert (the teacher) gave an introduction to OOP and the usefulness of the Abstraction. Also, he gave an explanation and clarification of how it could address the proposed exercise for a total understanding and ownership of the subject.

Students responded to a survey to gather their opinion of the VLE developed. In many cases it can be observed that the students preferred virtual education.

This final survey allowed to determine if any improvement had occurred. Results show that students expect and need more practical education, focusing on solving problems based on independent learning, but guided by both their teachers and monitors from higher semesters. This will surely allow teachers to understand the problems and difficulties that students go through and determine easy ways overcome this. As shown by the results to the question of evaluation of the tool:

- Do you think that virtual education is a useful alternative in reinforcing what you learned in class?

	Students	Percentage
Yes	22	100 %
No	0	0 %
Total	**22**	**100 %**

Table 3. Student's perception of virtual education as a good learning strategy

Another question was oriented to determine the students' perception of the project defined in the VLE:

- Do you think that developing a project helps to understand each OOP stage beginning with Abstraction?

	Students	Percentage
Yes	19	90.48 %
No	2	9.52 %
Total	**21**	**100 %**

Table 4. Student's perception of the project as a strategy to learn OOP.

Collaborative tools like wikis, chat, blog and forums designed to provide lasting support and collaborative construction must be accompanied by a strong motivation from the classroom by the teacher for its use and effectiveness, proposing discussions and promoting their value in the academic period, through the investigation of external material and answers to questions that students can observe. This conclusion is supported by the use the students found in their reflection when they responded the survey:

- Do you think that collaborative or cooperative work proposed in the forum, blog and wiki contributed to your learning?

	Students	Percentage
Yes	20	90.91 %
No	2	9.09 %
Total	**22**	**100 %**

Table 5. Student's perception of collaboration in VLE as a contribution to learn OOP.

It is convenient to update and modify the questions presented in the test periodically, so that students do not pass this information through the following groups that could make the tests invalid. As reflected in the student's responses, the learning platform is a key aspect of the successfulness of the implementation:

- Do you consider that the learning tool is easy to use (the platform)?

	Students	**Percentage**
Yes	22	100 %
No	0	0 %
Total	**22**	**100 %**

Table 6. Student's perception of the complexity of the developed platform.

It is important to mention that OOP is a difficult course for interdisciplinary students or students from fields other than computer science or similar programs.

6. Conclusions and future work

Virtual technology for learning is becoming a common strategy for new generation students. They are very positive to initiatives as the one proposed in this paper which not only makes it easier the implementation but is also a requirement for this strategy to be successful.

The VLE requires the commitment of the professor to update information, perform evaluations, and monitor the students' evolution in the platform. All that has to be connected with the activities performed in the face-to-face class. That is the reason why a blended strategy is recommended.

Approaching to a different learning strategy from the traditional learning concepts taught in class, contributes to enhance learning skills such as autonomy, critical thinking, self-motivation, among others. The VLE proposed in this paper constitutes an alternative to help students in their understanding of the subjects related to programming Object-oriented.

References

[1] Börstler Jürgen and Sharp Helen (2003). Learning and Teaching Objetc Technology. Special Issue, Journal of Computer Science Education,Vol 13.

[2] Bruce Christine, Buckingham Lawrence, Hynd John, McMahon Camille, Roggenkamp Mike, Stoodley Ian (2004) Ways of Experiencing the Act of Learning to Program: A Phenomenographic Study of Introductory. Transforming IT education: Promoting a culture of excellence. Informing Science Press, United States, p. 301-325.

[3] Heo Misook (2003) A Learning and Assessment Tool for Web-based Distributed Education. CITC4 '03 Proceedings of the 4th conference on Information technology curriculum. Florida State University, p.151 – 154

[4] Hwang Wu-Yuin et al (2008) A web-based programming learning environment to support cognitive development. Interacting with Computers. Vol 20, Issue 6, December 2008, p.524–534

[5] Jadud Matthew (2005) A First look at Novice Compilation Behaviour using BlueJ. Computer Science Education. Vol 15, p. 1-25.

[6] Keefe Karen et al (2006). Adopting XP Practices for Teaching Object Oriented Programming. Proceedings of the Eighth Australasian Computing Education Conference, Hobart, Australia.

[7] Kölling Michael (1999). The Problem of teaching object-oriented programming Part I, II: Languages. Journal of Object-Oriented Programming, Vol 11 (8), p. 8-15.

[8] Sitthiworachart Jirarat y Joy Mike (2003). Web-based Peer Assessment in Learning Computer Programming. ICALT: 180-184.

[9] Truong Nghi, Bancroft Peter, Roe Paul (2003) A Web Based Environment for Learning to Program. 26 Australasian Computer Science Conference, Vol 25, No 1.

[10] Vargas Germán, Gamboa Sarmiento, Sonia Cristina (2008) Didáctica en la condición postmoderna. De las competencias a la cooperación. http://dialnet.unirioja.es/servlet/dcfichero

[11] Villalobos Jorge (2007) Cómo enseñar a programar: un enfoque efectivo.

[12] Villalobos Jorge, Casallas Ruby, Marcos Katalina (2005) El reto de diseñar un Curso de Programación de Computadores. XIII Congreso Iberoamericano de Educación Superior en Computación, Pontificia Universidad Javeriana, p.48 - 55.

[13] Wei fang et al (2005) A student model for object-oriented design and Programming. Journal of Computing Sciences in Colleges. Vol 20 Issue 5, p. 260-273