Study Case of an Adaptive Educational Tool Oriented to University Students for an Object Orientation Course

Ninozka González, Claudio Cubillos(✉), Silvana Roncagliolo, and Rafael Mellado

Pontificia Universidad Católica de Valparaíso, Valparaíso, Chile
ninozka.93@gmail.com, {claudio.cubillos, silvana, rafael.mellado}@pucv.cl

Abstract. The use of e-learning for educational purposes has been the focus of researchers for a long time, because of the difficulty to determine if a tool based on this concept is able to instruct the educational content that a professor can easily teach on a classroom. In this document the available tools and teaching methods to deliver Adaptive Learning to each user are investigated. Through the development of this paper, the adaptive techniques to be implemented to the Educational Tool are proposed; how and which contents will be presented to the students and which adaptation model will be applied to that content. Thanks to the investigations on the topic, and the analysis of the performance of the students on the Object Orientation course, from the Pontificia Universidad Católica de Valparaíso (PUCV), preliminary models for the Educational and Adaptive Tool have been defined, with promising results.

Keywords: E-learning · Adaptive learning system · Educational tool · Adaptive Hypermedia System · Object-Oriented Programming · Java language

1 Introduction

Nowadays, the action of teaching tends to be related, in the first instance, to a classroom where a teacher instructs in a general and personalized way students who are interested in learning. But this process can not only be carried out depending solely on a teacher in a physical environment such as the classroom, we can also get teachings through, for example, computational online platforms.

In this work it is sought to demonstrate that the educational and adaptive tools are a plus in the classroom and that they support the study of students, improving their performance, offering more personalized content, including adaptation mechanisms. The research described in this document are done by students from the Pontificia Universidad Católica de Valparaíso (PUCV); the analyses were done to students who have taken an Object Oriented Programming course based on JAVA programming language, taught under an Informatics Engineering Bachelor Program.

This document is divided in sections; Sect. 2 the related and theoretical works are explained. In the third section the architecture of the adaptive tool used is explained.

G. Meiselwitz (Ed.): HCII 2019, LNCS 11579, pp. 153–169, 2019.
https://doi.org/10.1007/978-3-030-21905-5_12

In Sect. 4 a historical analysis of the students is done. In Sect. 5 the evaluation of the tool and the content to be used for testing. The tests done by the students are in Sect. 6. Finally, the conclusions can be found in Sect. 7.

2 Related Work

2.1 Educational and Adaptive E-learning Tools

Adaptive learning is creating a new approach to teaching action, where teachers must adapt existing methodologies in order to teach and students are changing their way of learning. Some of the educational and adaptive e-learning tools that are related to this work are the following:

- Ask-Elle: tutoring system used to teach Haskell programming language, by Gerdes et al., for the University of Utrecht in Norway [2]. The mechanism used in this work consists of verifying the correctness of incomplete programs and providing suggestions. The system makes it possible to deliver solutions with feedback messages.
- AtoL: is an intelligent tutoring system that dynamically adapts to the needs of each student and provides the student with immediate feedback. Realized by Yoo et al. [3].
- The JavaTutor System: It is a LE for Java with multimodal affection recognition. It takes into account the cognitive and affective aspects of students who use different hardware tools to recognize their affective state [15].
- CTutor: a problem-solving environment that diagnoses students' level of knowledge, but also provides feedback and advice to help them understand the course topic, overcome misconceptions, and reinforce concepts learned [12].
- JITS: Java Intelligent Tutoring System (JITS) involves the development of a programming tutor designed for students in their first Java programming course at university level. An overview of architectural design, artificial intelligence techniques, and the user interface is presented [13].
- J-LATTE: intelligent, constraint-based tutoring system that teaches a subset of the Java programming language. J-LATTE supports two modes: conceptual mode, in which the student designs the program without having to specify the content of the instructions, and coding mode, in which the student completes the code [14].

The mentioned systems have been made to support the teaching, most of these use artificial intelligence techniques, such as Intelligent Tutor Systems, to create student profiles and provide feedback to students.

However, some of the works listed above are aimed at teaching a programming language other than JAVA (as in Ask-Elle, CTutor and AtoL). Studies related to JAVA (such as in The JavaTutor System, JITS and J-LATTE) use adaptive mechanisms based on the student's learning styles, while in this work rules of adaptation are generated from profiles created from historical data extracted from students.

Besides, in these works the way of teaching is through the resolution of code problems giving feedback if a problem was solved in an inappropriate way; in the present work for each subject the theoretical contents are taught in a sequential way and in addition coding examples are presented. In addition, feedback is given to the students when they carry out the final tests of each module. Finally, there are few studies in the literature oriented to Latin America and university students, such as in the present work.

2.2 E-learning

The definition of e-learning, from the work done by Bowles [1], is explained as a novel approach to provide well-designed, student-centered, interactive learning environments that are available anytime, anywhere. The author of this book classifies the aspects of e-learning into the following:

- Pedagogical: referring to educational technology as a discipline of educational sciences associated to technological means, educational psychology and didactics.
- Technological: referring to information and communication technology, through the selection, design, customization, implementation, hosting and maintaining of solutions where proprietary open source technologies are integrated.

2.3 Adaptive Learning

Adaptability to a learner's personal interests, characteristics and objectives is a key challenge in e-learning. Adaptability is the ability to shape to the situation in which a given object is being subjected; in this case, adaptability means that learners are provided with a learning design that is adapted to their personal traits, interests and goals [4].

Personalization includes not only objectifying the student's styles, but monitoring the system's usage to adapt to the student's way of learning. Such systems can help students stay focused through patterns that adapt to changes [5, 6]. The system should perform, as far as possible, both the teacher's role and the construction of robust student models for each user, allowing:

- Adaptation in the study program of each user.
- Help in the navigation through the course activities.
- Support in the accomplishment of tasks, exercises and problem solving.
- Support resources at any time needed.

In Mathoff's work [16] a series of requirements are proposed which an educational system should have in order to manipulate the adaptive process, such as interactivity, adaptable instruction, robustness, direct control of the learning process, empirical evaluation; and to be friendly in use.

2.4 Intelligent Tutoring System

For Ovalle and Jiménez [8], Intelligent Tutoring Systems (ITS) aim to emulate the behavior of a human tutor. They are called "intelligent" to contrast them with traditional computer-assisted instruction systems, being its distinction the use of IT techniques such as Artificial Intelligence.

ITS can provide individualized education by adapting to each student's level of knowledge, learning abilities, and individual needs. These systems separate by modules the necessary components to form an ITS architecture:

- Domain Module: represents the knowledge that will be taught and pedagogically organized to ease the tasks of the tutor module.
- Tutor Module: is in charge of guiding the teaching-learning process.
- Student Module: represents the student's level of knowledge for the system.
- Educational Module: is responsible of the management of the interactions between the system and the users through the communication of the modules and the client.

2.5 Adaptive Hypermedia System

For the authors who carried out the research work presented in [7, 9], the Adaptive Hypermedia System (AHS) is capable of constructing a mock-up of the objectives, preferences and knowledge of each user, in order to use it dynamically through what is called a user model and a domain model. With this mock-up it is possible to adapt the content, navigation and interface to the user needs.

The overall architecture of an AHS must have three essential parts according to the works of Benyon [10] and De Bra [11] explained below:

- User model: describes the information, knowledge and preferences of the student.
- Domain model: provides a structure for the representation of user-dominated knowledge. This model stores the user's estimated level of knowledge for each concept defined in the course content.
- Interaction Model: represents and defines the interaction between the user and the application.

3 MAGLE Adaptive Tool Architecture

3.1 MAGLE Authoring Tool

For the presentation of contents, the MAGLE authoring tool was used, which allows to visualize the contents created in a web page, where each module and activity is represented. The acronym for the authoring tool used comes from Modular Adaptive and Gamified Learning Environment.

MAGLE is a learning management system for creating learning environments based on adaptation and gamification. It allows you to create e-learning content (lessons), organize courses, deliver content, register users in courses, and finally monitor and evaluate their performance. Generally speaking, we can say that it is:

- An online learning management software package.
- A virtual learning space aimed at facilitating the experience of distance training, both for educational institutions and companies, in mixed or semi-present form, and only virtual.

MAGLE allows you to create modules, clusters, and activities; each module can contain clusters and activities, and each cluster is a set of activities. The activities represent what a web page would look like. The tool allows you to enter layout, text, a series of types of exercises (such as alternative, multiple response and binary), and multimedia content, such as images or videos.

3.2 Teaching Material for the Adaptive Tool

For the tests, the tool covered the contents described in Table 1. These topics were chosen from the opinions of the same students carried out in the classroom, studies on the partial academic performance of the course and about the academic performance of the first formal evaluation.

Table 1. Contents covered by the tool.

Topics covered by the platform	
Topic 1	Classes
Topic 2	Objects, get and set methods, visibility modifiers
Topic 3	Overload
Topic 4	Collections

In the object orientation course, the first part of the content is described in the following Table 2.

Table 2. Contents covered for the first formal evaluation.

Content of the first part of the course	
Main topic	Sub-topic
Introduction	Origins
	Principles
	Languages
	Installation and compilation
Classes	Declaration
	Access
	Attributes
	Constructors
	Destructors
	Methods
	Control structures
	Data structures
GUI interfaces	Windows
	Events

3.3 Tool Content Architecture

The content architecture in the authoring tool is divided into topics, where each topic covers a specific content of the subject. The topics in turn have explanatory introductions, explanatory exercises and evaluative exercises; the evaluations that are carried out are Pre-test and Post-test. The module divisions and tool evaluations are explained below:

- Explanatory introduction: consists of an introduction to the content, which explains the theoretical concepts of the subject being studied, through plain text or images.
- Explanatory exercises: it consists of the realization of explanatory examples supporting the introduction of the topic.
- Evaluation exercises: it consists of a brief evaluation test where the contents presented in the module will be evaluated.
- Reinforcement: consists of a more detailed explanation and more examples of a particular subject.
- Pre-Test: consists of an evaluation of the student to know how much he handles the subject being studied.
- Post-Test: consists of an evaluation with the same questions as the pre-test but modified to compare how much the student learned with the tool.

Before starting with the explanation of the subject, and for study purposes, the students carry out a pre-test evaluation. After this, the students visualize the explanation of the topic, where it contains the exercises, and explanatory introductions. Once the student has gone through all the explanation of the subject of a specific topic or reinforcement a post-test is done. In this way, meaningful data will be obtained to compare student performance with the tool.

3.4 Tool Adaptation Model

Complementary explanations of sub-topics of each item were used for the adaptation model, leaving content visible for one user profile or another. Two user profiles were used, taken from the analysis made of the students; the adaptive explanatory contents are classified into two levels:

- Low: This grade is oriented to students who perform poorly, therefore the explanations consist of more examples and explanation slides.
- High: this grade is oriented to students who have good academic performance; the amount of explanation and examples is briefer than in the other case.

Using the MAGLE authoring tool, an adaptation is made as the student progresses through the course. This consists of, for each sub-topic of the main topic, evaluating the subject with a key question. If the student answers this question incorrectly, then for this student a deeper explanation will be shown with another type of exercise, so that he can understand why he made a mistake. If the student responds well to this question then the content will follow its normal path.

Below a flowchart to better explain how the adaptation mechanism works is presented in Fig. 1.

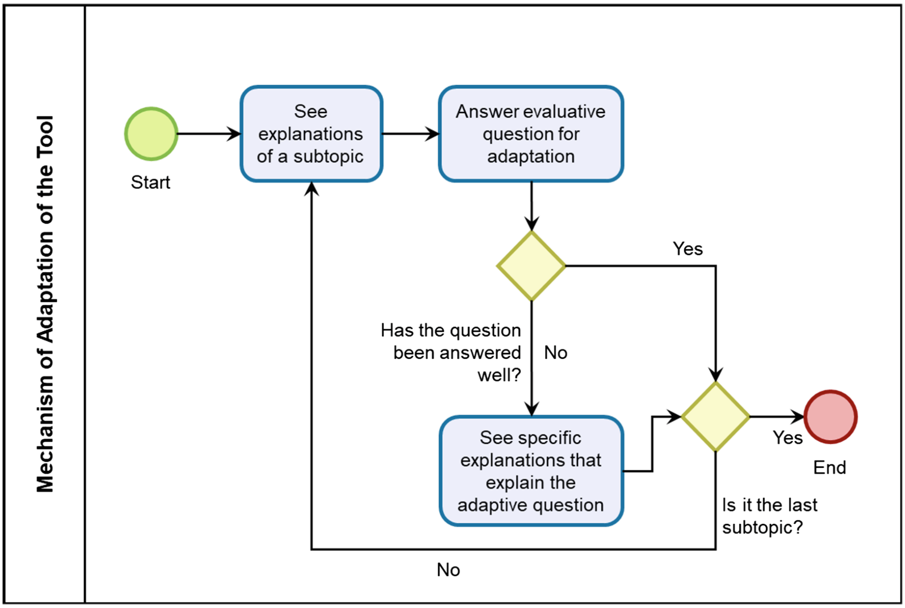

Fig. 1. Flowchart of adaptive model.

4 Stage 1: Historical Analysis of University Students to Obtain Adaptation Profiles

The analysis of the students was made with the objective of creating profiles and finding factors that influence the performance of the students who have taken the Object Orientation course, and that make them fail or pass the course. For this analysis, 107 university students from the PUCV who have already taken the Object-Orientation course between 2015 and 2016 were studied. Also, the studied variables were extracted from the academic web system used in the university (called Navegador Académico), where the students' grades are registered, in addition to the PSU scores.

4.1 Analysis of Academic Variables

Through the access to the Navegador Académico it has been possible to extract certain characteristics that at first sight have been intuited as possible factors in the approval or reprobation of the course. A total of eleven variables were initially considered (See Table 3).

Table 3. Variables extracted from the Navegador Académico.

Variable	Description
PG-S2	2nd semester average grade
PG-S3	3rd semester average grade
PG-S4	4th semester average grade
PG3-SA	Average grade of 3 semesters before target course
FG-ICI2240	Final grade in Data Structures course, pre-requirement of target course
FG-ICI1142	Final grade in Programming Fundamentals course, pre-requirement of the course Data Structure
PG-ICI2240-ICI1142	Average grade between Data Structure and Programming Fundamentals courses
%SR	Percentage of passed courses in the program until reaching the target course
SRI	Internal University Index that describes how risky is the student while staying in the career. The closer to 1, the riskier
PSU-PROM	Average PSU score when pre-enrolling in the career
PSU-POND	Weighed PSU score when pre-enrolling in the career

4.2 Results of the Analysis

Two variables were discarded, PSU-PROM and PSU-POND, due to their homogeneous behavior regarding the mean of the data (see Table 5), where the PSU-PROM variable has a mean of μ = 629 pts. and a standard deviation of σ = 40 pts; the PSU-POND variable has a mean of μ = 627 pts. and a standard deviation of σ = 37 pts.

Having a standard deviation in both variables very below the mean, it can be concluded that PSU variables are not very influential in generating student profiles, such that they fail or pass the Object Orientation course.

To verify the expressed conclusion with PSU variables, an analysis of all variables was performed with the Naive Bayes Multinomial algorithm, which is an algorithm used to predict independence between predictor variables. This algorithm is used to search each variable's weights and discover which factor is most significant for the failure or approval of the course of each student. The algorithm is applied with variables of the same type of measure, since it is possible to study the weight of the variables over others. They were grouped as follows:

- Percentage Variables: Such as the percentage of assed courses and the SRI.
- PSU Variables: Such as the weighted PSU and the average PSU.

Note that scale variables from 1.0 to 7.0: Such as those that are the student's final or average grade. Exit classifications are named as:

- "CLASS 1": represents the course's failure.
- "CLASS 2": represents the course's approval.

Table 4. Weights of the percentual variables

Variables	Class 1	Class 2
%SR	0.55	0.89
SRI	0.44	0.1

In the next table the PSU variables are analyzed (see Table 5), being able to notice that they do not influence much one variable over the other in approving or failing the course.

Table 5. Weights of the PSU variables

Variables	Class 1	Class 2
PSU-POND	0.5	0.5
PSU-PROM	0.49	0.49

Finally, with the variables in the grading scale, it was noted that the grades of the course ICI2240, ICI1142 and the average of both have more weight over the other variables analyzed. It was also noted that in the semester that the course ICI2240 was dictated, it also has more weight than the other variables in the approval or failure of the course (see Table 6).

Table 6. Weights of the note variables

Variables	Class 1	Class 2
PG3-SA	0.137	0.138
FG-ICI2240	**0.16**	**0.16**
FG-ICI1142	**0.14**	**0.142**
PG-ICI2240-ICI1142	**0.153**	**0.155**
PG-S2	0.134	0.125
PG-S3	**0.137**	**0.141**
PG-S4	0.136	0.135

The first variables analyzed were the percentual ones, as it can be seen in the following Table (see Table 4), the variable of %SR influences much more in the approval of the course than the variable of SRI, while for the course failure %SR also has more weight than the SRI.

To develop a predicting algorithm the most significant variables were chosen (PG-ICI2240, FG-ICI1142, PG-ICI2240-ICI1142, %SR and SRI). The resulting model obtained a 81% successful classification, but in this case, the most important thing is to find out if the algorithm successfully classifies the students in failed profile rather than approval. For this profile, the algorithm only achieved 65% success in the ranking. Table 7 describes the error percentage of the algorithm (See Table 7).

Table 7. Percentage of error and classification

	% of error
Error	19%
Class 2 classification	35%
Class 1 classification	0%

Given these results, we can conclude that there is a relationship between students who, during the course of the career, have had low grades in general and in the prerequisite courses of the Objects Orientation course. That is to say, it is much more likely that this profile of students will fail the course, compared to the profile of students with good grades.

5 Stage 2: Test for the Tool and Content Evaluation

Prior to testing the adaptive tool with students, a survey was conducted to randomly selected students to evaluate the design of the contents that was presented to students and the proper functioning of the MAGLE authoring tool.

5.1 Content Evaluation by the Students

For these tests, students from the School of Computer Engineering at the PUCV participated, with a total of 13 people. The assessments of the students for the explanation of the content were not good, of a total of 5 questions made with the Likert scale, which evaluated the presentation of the content, 4 of them were rated as deficient. Given these results, the presented content was changed, more examples were added and the way of explaining the theory was reformulated, so that it would be more didactic and simpler for the students. The changes made were validated with the course professors. The following Table 8 shows the results of the evaluation done by the students, separated by item.

Table 8. Students' assessment regarding the content

Variables	Strongly disagree	Disagree	Neutral	Agree	Strongly agree	Total
The content explanation was clear	0	1	8	1	3	13
I agree with the amount of examples taught in the tool	0	4	8	1	0	13
I agree with the level of deepness of the content	0	3	9	1	0	13

(*continued*)

Table 8. (*continued*)

Variables	Strongly disagree	Disagree	Neutral	Agree	Strongly agree	Total
The content shown in the tool proved to be didactic	0	10	2	1	0	13
The content presentation motivated me to continue studying it	2	3	7	1	0	13
Total	2	21	34	5	3	

Table 9. Evaluation of students regarding the tool MAGLE

Variables	Strongly disagree	Disagree	Neutral	Agree	Strongly agree	Total
The tool follows a normal flow, without connection errors, throughout the process	0	0	0	2	11	13
It did not take me long to get to the target when I was navigating through the tool	0	0	0	8	5	13
User registration of the tool is quick and simple	0	0	0	0	13	13
The tool contains easy-to-access buttons and instruction for the user	0	0	1	2	10	13
I was able to complete my task without visual problems	0	0	4	5	4	13
Total	0	0	5	17	43	

5.2 Evaluation by Students of the MAGLE Tool

In general, the ratings regarding the tool were very good, the navigation in it was measured, its performance and user-friendly interface. The students evaluated the following items regarding the tool (see Table 9).

6 Stage 3: Student Tests with the Tool

For the following tests, students were selected from a Database course of the career in IT Engineer ("Ingeniería en Ejecución en Informática"), because these students have already passed a Data Structure course and have not had an Object-Oriented Programming course yet, therefore the first contents will be better evaluated than with students who had already done the course. For reasons of better explanation of contents, the following topics were selected:

- Topic 1: Classes and their components.
- Topic 2: Visibility Modifiers and Methods.

6.1 Content Testing Topics 1 and 2, with the Tool Without Adaptation

For these tests, the students were given a pre-test and post-test to evaluate how much help the educational tool provided to the students. The results were as follows (See Table 10).

We concentrated on topic 2, specifically because the students answered all of the test questions. As seen in Table 10, students have a considerable improvement of at least 25% when they study the content with the tool.

For analysis purposes, two biases were performed in the pre-test row of topic 2, when students were not yet passing through the tool explanations; the first bias was taken with the measurement of the mean of 4 good questions; the second bias was taken with the measurement of the mean of 3 good questions. With the objective of comparing the performance of the students who had lower grades, with the students who did the pre-test without problems; thus obtaining 4 groups:

- Bias from minor to 4 good questions.
- Bias from greater than 4 good questions.
- Bias from minor to 3 good questions.
- Bias from greater than 3 good questions.

The following table (see Table 11) shows the results of this bias, calculating the mean and standard deviation in the groups.

Table 10. Results table for 1st and 2nd topics using tool without adaptation

Student	Pre test 1st topic	Post test 1st topic	Pre test 2st topic	Post test 2st topic	PG3	% SR	Grade: Prog. fundamentals	Grade: Data structures	Course average	Pretest and posttest delta value	Improvement percentage
1	6	6	2	6	4,1	69	5,1	3,9	4,5	4	50
2	5	6	3	6	5,1	75	6	1,5	3,8	3	37,5
3	7	7	3	6	5,3	100	4,8	4,5	4,7	3	37,5
4	5	5	3	8	5	98	5,1	4,2	4,7	5	62,5
5	5	7	4	8	3,8	64	4,2	4,9	4,6	4	50
6	6	8	4	6	5,1	100	5,7	4,7	5,2	2	25

(*continued*)

Table 10. (*continued*)

Student	Pre test 1st topic	Post test 1st topic	Pre test 2st topic	Post test 2st topic	PG3	% SR	Grade: Prog. fundamentals	Grade: Data structures	Course average	Pretest and posttest delta value	Improvement percentage
7	6	6	4	7	4,7	70	4,1	4,1	4,1	3	37,5
8		6	4	6	4,6	75	4,2	6	5,1	2	25
9	8	7	4	8	5,5	100	5,9	5,8	5,9	4	50
10	8	5	5	8	5,4	100	5,1	6,1	5,6	3	37,5
11	6	7	5	6	4,3	67	5,2	4,8	5	1	12,5
12	7	6	6	8	6,5	100				2	25
13		6	6	8	5,1	98	6,4	6,2	6,3	2	25
Mean	6,1	6,2	3,7	6,9	4,9	83,1	5,0	4,1	4,9	2,5	
Standard deviation	1,1	0,9	1,2	1,0	0,7	15,5	0,8	1,3	0,7	1,1	

With these results we can conclude: that the students who have lower grades in the pre-test and pass through the explanation of the tool, improve considerably their results and have a higher delta of improvement than the students who have more knowledge and pass through the tool. With this we can affirm that the students who have lower grades can reach the students who have higher grades; minimizing the performance difference among the students.

Table 11. Analysis of biases of topic 1 and 2 with the tool without adaptation

Bias	Measure	Pre-test topic 4	Post-test topic 4	PG3	% SR	Grade: Programming fundamentals	Grade “Estructura”	Course average	Pre and post test delta
Bias minor to 4 good questions	**Mean**	3,4	6,7	4,8	83	5	4,4	4,7	**3**
	Standard deviation	0,7	0,9	0,5	16	0,7	1,3	0,6	**1**
Bias greater than 4 good questions	**Mean**	5,5	7,5	5,3	91	5,6	5,7	5,6	**2**
	Standard deviation	0,6	1	0,9	16	0,7	0,8	0,7	**0,8**
Bias minor to 3 good questions	**Mean**	2,8	6,5	4,9	86	5,3	3,5	4,4	**4**
	Standard deviation	0,5	1	5,3	16	0,5	1,4	0,4	**1**
Bias greater than 3 good questions	**Mean**	4,7	7,2	5	86	5,1	5,3	5,2	**3**
	Standard deviation	0,9	1	0,8	16	0,9	0,8	0,7	**1**

6.2 Content Testing Topic 1 and 2, with the Tool Including the Adaptation Mechanism

The test that was carried out with the adaptation mechanism had a total number of 16 questions; there were also 12 reinforcement questions that were in charge of the adaptation mechanism. The results of these tests with the tool and the adaptation mechanism can be seen in Table 12.

Table 12. Results table for 1st and 2nd topics using tool with adaptation

Student	Time: Content	Time: Final test	Reinforcement	Good answers	% SR	PG-3S	Grade: "Data structure"	Grade: "Informatics introduction"	% Good answers
1	22	8	**1**	12	100	6,1	–	–	75
2	16	8	**1**	14	100	6,1	6,9	6,6	87,5
3	22	8	**1**	14	86	5,1	5,8	6	87,5
4	14	8	**1**	16	100	6,1	6,7	6,4	100
5	10	8	**2**	12	100	5,5	5,8	5,9	75
6	22	10	**2**	14	62	4,9	5,8	4,7	87,5
7	20	8	**2**	16	98	5,1	6,2	6,4	100
8	16	8	**4**	10	100	5,5	5,5	5	62,5
9	18	12	**4**	14	80	4,6	4,3	5,3	87,5
10	22	10	**4**	14	100	5,3	4,6	5,6	87,5
11	20	8	**4**	16	100	5,3	4,5	4,8	100
12	14	8	**4**	16	98	5,5	4,7	6,5	100
13	16	8	**6**	8	69	4,3	4,1	4	50
14	16	12	**6**	12	100	5,1	4,7	5,7	75
15	20	12	**7**	12	48	3,6	4,1	3,9	75
16	20	8	**7**	14	98	5.3	5,5	6,3	87,5
17	20	8	**7**	14	76	4,2	4,3	5	87,5
18	24	12	**9**	14	100	5,5	4,6	5,6	87,5
19	20	10	**9**	14	71	4	3,3	3,6	87,5
20	20	12	**10**	12	69	4,1	3,9	5,1	75
Mean	**19**	**9**	**5**	**13**	**88**	**5,1**	**5,1**	**5,4**	
Standard deviation	**4**	**2**	**3**	**2**	**16**	**0,7**	**1**	**0,9**	

As in previous tests, for reasons of analysis, two biases focused on the row of number of reinforcement used by the student were carried out. Obtaining four groups in this way:

- Bias of 6 or more reinforcement questions.
- Bias less than 6 reinforcement questions.
- Bias of 7 or more reinforcement questions.
- Bias less than 7 reinforcement questions.

These biases have the purpose of compare between students who required more reinforcement to understand the subject and the ones who did not require reinforcement. The results can be seen in the following Table 13.

Table 13. Analysis of biases of topic 1 and 2 using the tool with adaptation

Bias	Measure	Reinforcements	Good answers	% SRI	PG3	Grade "Data structure"	Grade "Informatics introduction"
Bias of 6 or more reinforcement questions	**Mean**	6,2	**13**	85	4,8	4,5	5,1
	Standard deviation	2,2	**2,3**	17	0,7	0,6	0,8
Bias less than 6 reinforcement questions	**Mean**	1,4	**14**	92	5,6	6,2	6
	Standard deviation	0,5	**1,6**	14	5,4	0,5	0,7
Bias of 7 or more reinforcement questions	**Mean**	7,6	**13**	78	4,5	4,3	4,9
	Standard deviation	1,5	**2,1**	19	0,7	0,7	1
Bias less than 7 reinforcement questions	**Mean**	2,5	**14**	94	5,4	5,5	5,8
	Standard deviation	1,4	**1,9**	11	0,5	1	0,7

We can conclude that students who score lower on pre-test tests and see the explanation of the tool considerably improve their scores and have a much higher delta than students who already have the knowledge and go through the tool. With this we can affirm that the students who have bad grades can reach the students who have good grades; minimizing the gap between groups of students.

6.3 Analysis of Results

If we rank students according to the average grade of the prerequisites courses such as Data Structure and Programming Fundamentals we can draw the following conclusions from the tests performed:

- The percentage of courses approved by the students during the university career is related to the performance of the students in these tests.
- Students with grade averages lower than 5.0 in the prerequisite courses had to go through more reinforcement content than students with grades higher than 5.0.
- The time spent on the test by students with grade averages below 5.0 is greater than those with grade point averages above 5.0.
- The percentage of improvement for the two tests increases considerably when studying the contents through the educational tool.
- In both tests, the amount of good answers in both biases does not have much variation, when students already pass through the tool.
- The reinforcement used in the content turned out to be satisfactory, since the students who had to go through a reinforcement were leveled with the students who didn't have to go through a reinforcement.

7 Conclusion

An adaptive e-learning system was developed and novel adaptation mechanism was implemented based in the profiling of students based on its academic performance in previous courses. Based on our previous studies of the educational topic [6, 17–19] together with the opportunity to work directly in the classroom, gave a plus in being able to leverage a solution capable of adaptively teaching and reinforcing topics on students.

The preliminary study of the factors that influence the approval or disapproval of the course has been able to reflect certain variables of great weight for the students, and provides information necessary to focus the tool on those students with the highest risk of failing the course.

In this document, the good acceptance of e-learning tools in students has been made known and a positive impact on the learning process is reflected, together with an adaptation mechanism that supports and generates a 'leveling factor' in the knowledge of the participating students of this experiment. With the tests carried out in this work we left open the possibility of further research and experiments on university students, either in the same course or with other courses and topics since the MAGLE tool allows to modify the contents and the mechanisms of adaptation.

References

1. Bowles, M.S.: Relearning to E-learn: Strategies for Electronic Learning and Knowledge. Melbourne University Press, Carlton (2004)
2. Gerdes, A., Heeren, B., Jeuring, J., van Binsbergen, L.T.: Ask-Elle: an adaptable programming tutor for Haskell giving automated feedback. Int. J. Artif. Intell. Educ. **27** (1), 65–100 (2017)
3. Yoo, J., Pettey, C., Yoo, S., Hankins, J., Li, C., Seo, S.: Intelligent tutoring system for CS-I and II laboratory. In: The Annual ACM Southeast Conference 2006, pp. 146–151 (2006)
4. Van Rosmalen, P., Brouns, F., Tattersall, C., Vogten, H., Van Bruggen, J., Koper, R.: Towards an open framework for adaptive, agent-supported e-learning. Int. J. Continuing Eng. Educ. Life-Long Learn. **15**, 261–275 (2005)
5. Rani, M., Nayak, R., Vyas, O.P.: A ontology-based adaptive personalized e-learning system assisted by software agents on cloud storage. Knowl.-Based Syst. **90**, 33–48 (2015)
6. Diaz, F., Cubillos, C., Mellado, R., Barbaguelatta, E.: Development of a prototype of e-learning based on ontologies to analyze the impact of learning styles on engineering students. In: Proceedings of the 36th International Conference of the Chilean Computer Science Society, SCCC 2017, Arica, Chile, pp. 1–9 (2017)
7. Brusilovsky, P.: Methods and techniques of adaptive hypermedia. User Model. User-Adapt. Interact. **6**, 87–129 (1996)
8. Ovalle, D.A., Jimenez, J.A.: Ambiente inteligente distribuido de aprendizaje: Integración de ITS y CSCL por medio de agentes pedagógicos. Revista EIA **6**, 89–104 (2006)
9. Martins, A.C., Faria, L., Vaz de Carvalho, C., Carrapatoso, E.: User modeling in adaptive hypermedia educational systems. Educ. Technol. Soc. **11**, 194–207 (2008)
10. Benyon, D.: Adaptive systems: a solution to usability problems. User Model. User-Adapt. Interact. **3**, 65–87 (1993)

11. De Bra, P.M.E., Aroyo, L.M., Chepegin, V.: The next big thing: adaptive web-based systems. J. Digit. Inf. **5** (2004)
12. Kose, U., Deperlioglu, O.: Intelligent learning environments within blended learning for ensuring effective C programming course. Int. J. Artif. Intell. Appl. **3** (2012)
13. Sykes, E.R., Franek, F.: An intelligent tutoring system prototype for learning to program Java TM. In: Proceedings of the 3rd IEEE International Conference on Advanced Learning Technologies (ICALT03) (2003)
14. Holland, J., Mitrovic, A., Martin, B.: J-LATTE: a constraint-based tutor for Java. In: 17th International Conference on Computers in Education 2009, pp. 142–146 (2009)
15. Wiggins, J.B., et al.: JavaTutor: an intelligent tutoring system that adapts to cognitive and affective states during computer programming. In: Proceedings of the 46th ACM Technical Symposium on Computer Science Education, p. 599 (2015)
16. Masthoff, J., van Hoe, R.: APPEAL: a multi-agent approach to interactive learning environments. In: Perram, J.W., Müller, J.-P. (eds.) MAAMAW 1994. LNCS, vol. 1069, pp. 77–89. Springer, Heidelberg (1996). https://doi.org/10.1007/3-540-61157-6_23
17. Bourgeois, M., Sentis, F., Cubillos, C., Mellado, R., Roncagliolo, S.: Ashy.alRescate(): a videogame for developing basic object oriented programming skills. In: Proceedings of the 37th International Conference of the Chilean Computer Science Society, SCCC 2018, Santiago, Chile (2018)
18. Silva, R.M., Cubillos, C., Melgarejo, B., Roncagliolo, S., Velasquez, C.: Role videogame tool for teaching myths and legends to primary school students. In: Proceedings of the 37th International Conference of the Chilean Computer Science Society, SCCC 2018, Santiago, Chile (2018)
19. Silva, R.M., Barbaguelatta, E., Cubillos, C., Diaz, F.: X9: an adaptive learning platform for geometry at school level. In: Proceedings of the 37th International Conference of the Chilean Computer Science Society, SCCC 2018, Santiago, Chile (2018)