

A Didactic E-learning Platform with Open Content Navigation and Adaptive Exercises

Jonatan Gómez, Elizabeth Leon, Arles Rodriguez, Edwin Camilo Cubides

Julian Mahecha, Juan Camilo Rubiano, William Prado

Departamento de Ingeniería de Sistemas e Industrial

Facultad de Ingeniería

Universidad Nacional de Colombia

Bogotá, Colombia

{jgomezpe, eleonguz, aerodriguezp, eccubidesg, jrmahechad, jcrubianoz, wapradol}@unal.edu.co



ABSTRACT: *In this paper an intelligent platform of virtual learning is presented. The platform supports autonomous learning of students and allows administrators to develop virtual academic programs. Platform has tools for classes and quizzes and generates a map of contents based on the relationship between content units allowing students to navigate the topic in an autonomous way. Platform can represent the relationship between courses in a career, in this case Systems Engineering at Universidad Nacional de Colombia. A course of basic programming and a tool developed to learn functional programming called *fp – Learning* are also presented.*

Keywords: E-Learning, Virtual Learning, Intelligent Learning, Content Navigation

Received: 1 March 2012, Revised 18 April 2012, Accepted 23 April 2012

© 2012 DLINE. All rights reserved

1. Introduction

In the years (thanks to Computing and Communications technologies), educational models have been transformed from monolithic lineal models to flexible non lineal models centred in the skills and abilities of each student. Such flexible and non-lineal model allows students to learn in their own way and at their own rhythm [1] [2] [3] [4] [5].

In this transformation, several different educational process, from class room attendance education process to distance education process (using radio, TV, internet and mobile devices), have been included [6].

E-learning is a distance educational model which integrates information technologies with pedagogical elements in order to teach and train “*on line*” students in a particular topic [1]-[5]. An e-learning system is composed by several subsystems including but not limited to the following sub-systems:

1) Learning Management System [7] [8] [9]: In charge of managing, distributing and controlling the training activities. In particular, this sub-system manages users accounts and learning resources (documents, activities, assignments, etc), follows the learning process of each student, applies test, generates individual reports, and manages communication resources such as chat, video conferences and blogs.

- 2) Content Management System [10] [11] [12]: In charge of defining and maintaining the educational program content (usually courses content). It is done by defining content units with the possibility of being managed, personalized and used at different moments of the learning process.
- 3) Intelligent Tutoring System [13] [14] [15]: In charge of guiding the student, in an interactive manner, through the educational program content by using feedback, i.e., according to the perception the tutor systems has about the student.
- 4) Computer Adaptive Test [16] [17] [18]: In charge of maintaining and applying adaptive tests to students. Here, an adaptive test is a test that is able to change questions, according to the answers initially provided by the student. In this way, the test provides a better characterization of the level reached by the student, his/her difficulties and achievements.
- 5) Topic interaction and simulation tools: A set of computational elements, such as programming compilers, virtual laboratories, painting tools, CAD engines, etc., that are integrated in the e-learning system, in order to allow students to apply or test their learned skills and abilities.

In particular if contents, tests, and tutors are able to adapt according to the skills of each student, the e-learning system is called adaptive e-learning system [19]. The purpose of this paper is to show the progress made by Universidad Nacional de Colombia (National University of Colombia) in the definition and implementation of an adaptive e-learning system. Basically, the Content Management System is defined and implemented as a directed graph (non-linear structure) of content units. Each content unit is defined as a set of short videos (each video with an associated question, examples, exercises and written document), adaptive tests, and intelligent tutor system. The content graph is presented as an interactive tool allowing the student to go to any content through an special node called the “**HUB**” but also following the recommendation given by the content designer.

This paper is organized as follows. In Section II we present the module classes and the tools to interact with students. Section III we present the map module, the window where student explores the content units and the base component (**HUB**). Section IV we present the algorithm to locate the position of the content units and the **HUB**. Section V we present two options of the language interface of the platform, one in English and another in Spanish. Section VI we present the platform data model used for storing all information about students, courses, etc. Section VII we present a proposal of a course of functional programming, this content and the application *fp-learning*. Section VIII we present the traditional platforms used for teaching at the Universidad Nacional de Colombia. Finally, we present some conclusions.

2. The Classes Module

The classes module is inspired in the structure proposed by Stanford University in some on-line courses offered since the second term of 2011 (i.e. artificial intelligent course [20]). In this structure each class includes a series of videos and quizzes prepared by the professor. Key concepts of a content unit are presented in short videos with a duration no longer than 10 minutes. If the videos are too long students could get bored. After some of the videos the student has to present some questions related to the video. If the answers were wrong a video with an explanation about the right answer is presented.

A similar structure were developed (Figure 1); the current version of the platform includes a series of videos with his previews, in this way the student can select which video want to see, as shown in Figure 2. Quizzes are under development, the quizzes could have multiple choices with one or multiple answers, or a simple numeric answer in case the quiz requires do some little exercise.

Also the class module include a chat module (see Figure 1 right); in which students in the same class can discuss about the videos, if some of them has a question or comment other could help them. The chat allow send message to everyone in the class (public messages) or and specific user (private messages). Private messages are sent using @username, this is based on the style used by social networks such as facebook or twitter. Chat create a better experience for the students while they are using the platform, making them feel as if they were in a real class talking with their classmates, a monitor or a professor.

The class interface has a navigation bar below which presents the possibility to go to the course map (home page), the exercises and documents of the class (these two modules are under development) see Figure 1 (bottom). Exercises will be classified according their difficulty (this classification is made by the professor); the idea is that the platform adapts the exercises to the student. If a student can solve an exercise, the platform suggest a harder exercise than the previous one. If a student cannot solve the exercise, the next one will be easier.



Figure 1. Class view



Figure 2. Video view and pre-views

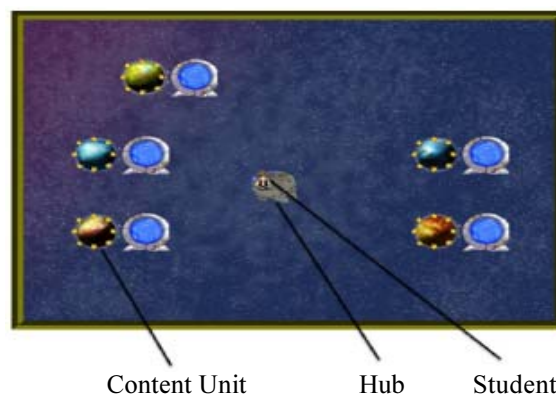


Figure 3. Contents map

The system will keep track about topics done, videos watched and quizzes solved by a student. With this data the system could suggest to the user which topic will be next, based on the performance of all the students.

3. Map Module

Map is a graphical representation of the relationship between content units. Map allows students to explore the content units that belong to a course. Main idea is to give a student the freedom to choose the topic of his/her preference. The base component of the map is the “**HUB**”. **HUB** is defined as a point in the space in which the student starts. Student can go from this point to a content unit. Each content unit is represented by a planet. If a content unit is available from the gate, the planet is surrounded by stars. In Figure 3, all the topics are available from the gate.

By clicking a planet the student is teleported to that place and a Dialog with options for classes, exercises and documents about a topic are displayed (Figure 4). Each content unit has a teleport that allow the student to go to another dependent unit or to the **HUB**. When a topic is completed a flag is displayed (Figure 5).

4. Algorithm to Locate Content Units Automatically

An algorithm to locate the contents map is proposed in this section, the position of each content units is generated from a directed graph, where the units are its nodes and relationships are its edges. To show an example of this algorithm in Figure 6 we presents an academic curriculum of the career of Systems Engineering of the Universidad Nacional de Colombia which is a directed graph; each label of the graph is replaced by a number. Graph is read from top to bottom and left to right, thus MATEMÁTICAS I is replacement by ①, EXPRESIÓN GRÁFICA is replacement by ②, and so on until MATEMÁTICAS II is replacement by ⑥, FÍSICA I is replacement by ⑦, the same happens with the rest of the 56 nodes. The edges among nodes are defined as sorted pairs such as: $\{(\textcircled{1}, \textcircled{6}), (\textcircled{1}, \textcircled{7}), (\textcircled{6}, \textcircled{12}), (\textcircled{6}, \textcircled{13}), (\textcircled{6}, \textcircled{29}), (\textcircled{7}, \textcircled{13}), (\textcircled{12}, \textcircled{18}), (\textcircled{12}, \textcircled{19}), (\textcircled{12}, \textcircled{20}), (\textcircled{12}, \textcircled{25}), \dots\}$.



Figure 4. Content unit options

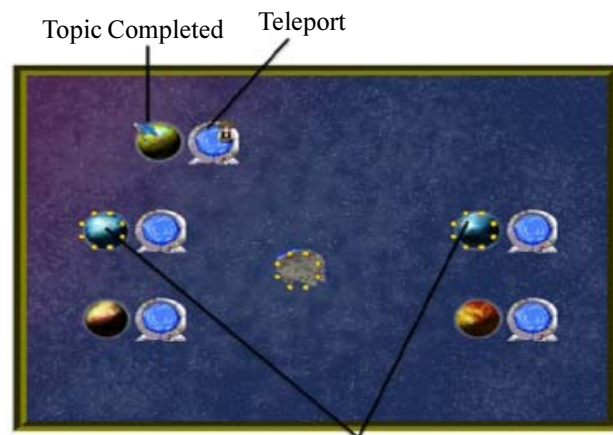


Figure 5. Content navigation

Grid is created from the set of edges and the number of nodes in the following way: i) *Isolate nodes* are searched, these are nodes without descendants nor ancestors together with the special node **HUB**. ii) *Initial nodes* are searched and stored in a list, these are nodes with descendants but without ancestors; this list is stored as first element in another list L . iii) Nodes which are second component in the edges where the first component are initial nodes are searched and stored in a list; this list is stored as second element in the list L . The same process is done with the rest of nodes which are second component in the edges where first component in the edges set is in the last list of the list L , and so on, until complete the search with all nodes. iv) As a node may be repeated in two distinct lists, all occurrences are deleted except the last. v) From first list into the list each node is located on Map from top to bottom separated by two cells. After that, two cell are moved towards right and each node into second list into the list is located from top to bottom, and so on with the rest of the lists into the list. vi) **HUB** node is located two cells below the previous nodes at the center of the grid. vii) isolate nodes are located two cell below of the **HUB** node.

When a new node is add to set of nodes, it is located after of the set of isolated nodes. Figure 7 shows a part of the grid obtained from last directed graph (curriculum of Figure 6) and application of previous algorithm.

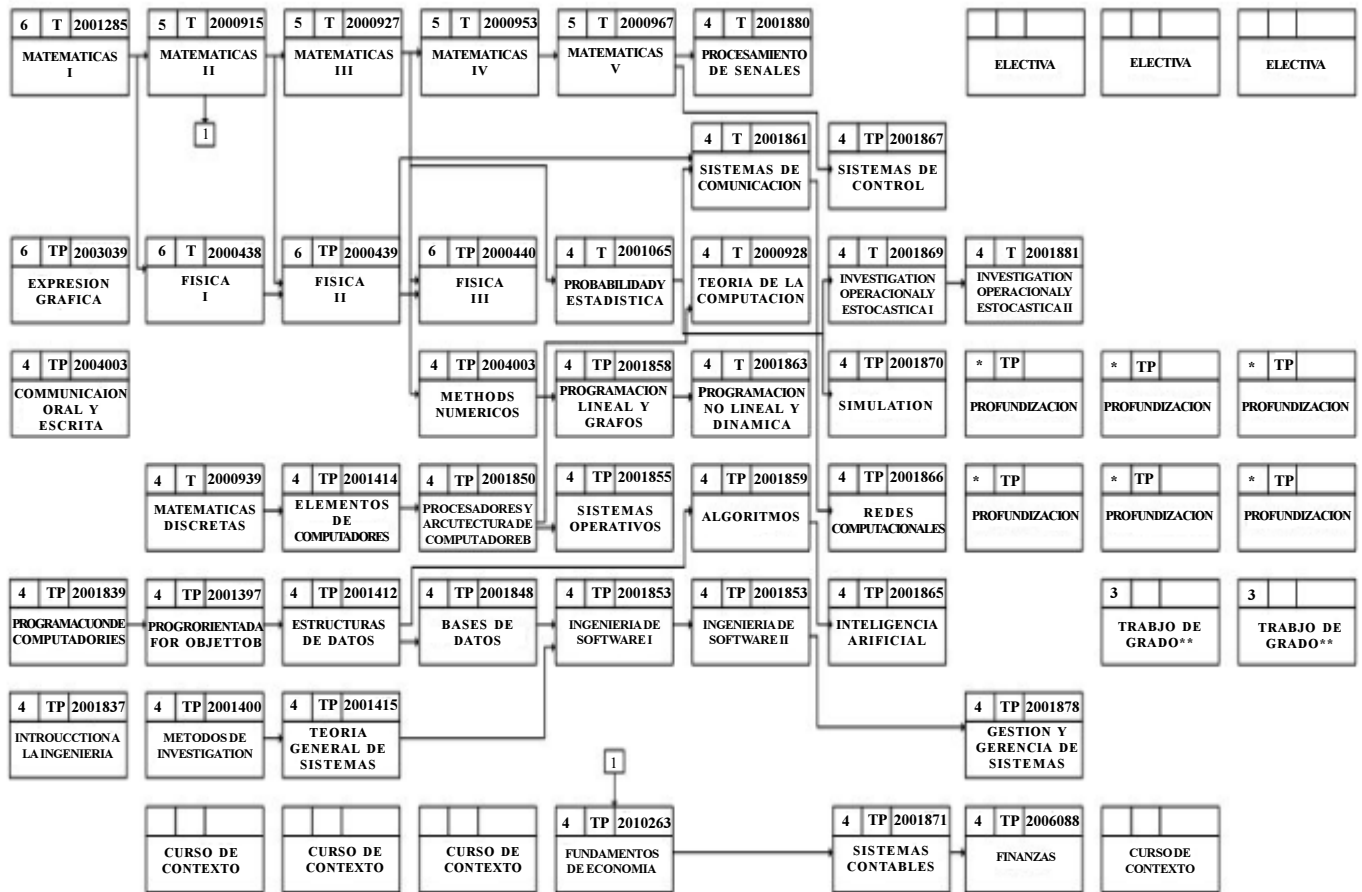


Figure 6. Example of a directed graph for initializing position on a Map



Figure 7. Grid obtained from graph presented in Figure 6 and the use of algorithm to automatically locate content units on the Map Module

5. Internationalization

One of the key objectives of the platform is to be viewable in different languages, in order to guarantee that no matter where a teacher or student comes from, or simply which language preference user has, they will always be able to work with the platform.

To achieve this, the platform has been designed to work with the user's preferred language, according to the language they have set in their browser. However, in case the user wishes to change the language the platform is displayed in, this is possible just by clicking a button on the site. An example of this is shown in Figures 8 and 9.



Figure 8. The platform viewed in English language

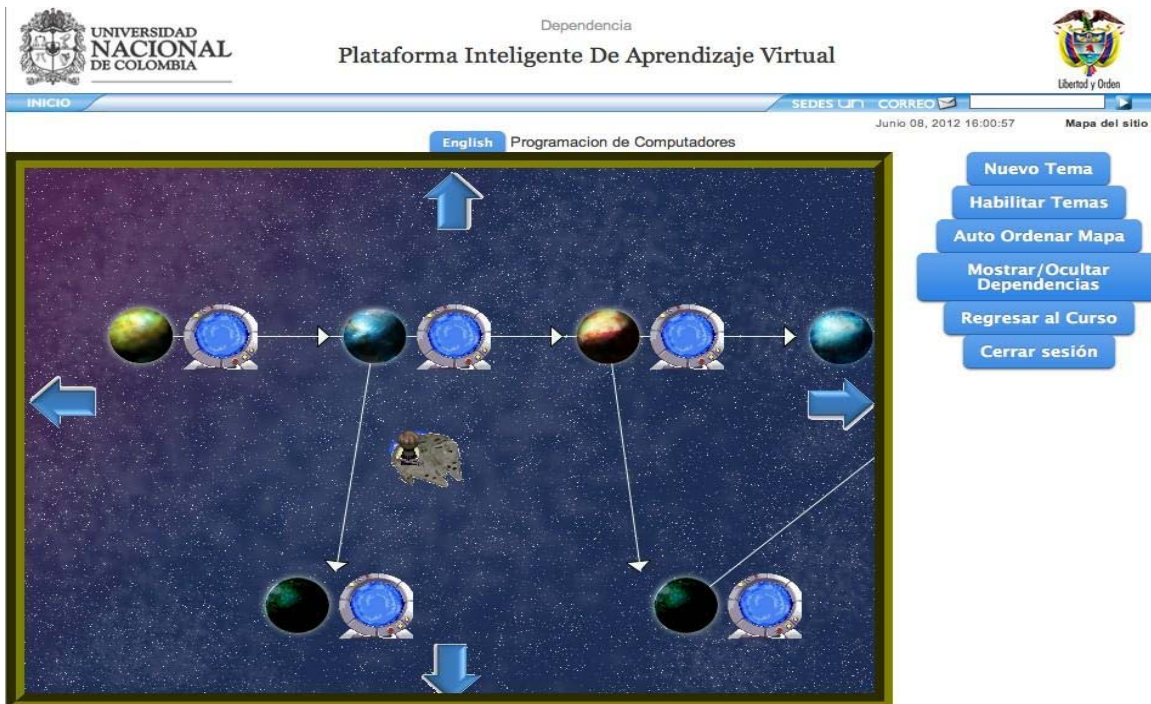


Figure 9. The platform viewed in Spanish language

Internationalization takes advantage of the *Gettext* php module [21] for translations, and *Smarty* [22] to keep the site's text content loosely coupled from the business logic code, which also makes it more easily extensible. This works in such a way that the translations of the menus are located in dictionaries, one for each language, placed in a separate folder, inside the *locale* folder. So, in order to add support for a new language, all that's required is to add a new folder with the appropriate language dictionary file inside it. This dictionary files can be easily created by employing a gettext catalog editor, such as *Poedit* [23].

6. Data Model

The platform can store all the information about the users, data like personal information, registered courses, and history about viewed courses and grades. In the same way database store all the information about existing courses in the platform, keeping the relationship between courses, like dependences with other courses and the unit contents of each one of them. Although the platform allows a free navigation in the different content units for each course, without restrictions, the professors that create the course can establish a dependence notion to guide the student in what they believe is the best way to take the course. Also is stored information about videos, exercises and documents associated to each topic, in this way all the important information to the student will be available in the moment that the student need it.

In the same way, all the information about exercises and test developed for each student, like responses, time, among other; is stored with the purpose of adapt the course according to the student progress. Also logs about navigation in the platform and chat conversations are stored, to make data mining over this information and determine the best way to advise the student or to correct the design of the course.

7. Application: Course of Functional Programming

A course of Functional Programming is proposed to be the first course to teach using this e-learning platform as pilot test. It is a basic course of computer programming, the idea is to teach both structural programming and declarative programming. Declarative programming is focused on functional programming, to teach this kind of programming a system called *fp-Learning* was designed. *fp-Learning* allows students to program with a functional style using Peano Arithmetic the primitive value “0”, the successor function, a list constructor and recursive functions. This system has mainly three modules: a module for interpreting functional programs (see Figure 10) [24], for inducing functional programs [25], [26], [27], [28], [29] and a debugger of functional programs [30], [31], [32], [33], [34], [35].

The content units of this course are:

- 1) Syntax of *fp-Learning* (terminals and functions).
- 2) Logic operations (and, or, xor, etc.).
- 3) Arithmetic operations (sum, product, factorial, logarithm, etc.).
- 4) List operations (length, append, inverse, etc.).
- 5) Sorting algorithms (Bubble sort, Insertion sort, Merge sort, Selection sort, Quicksort, etc.).
- 6) Additional recursive algorithms (Tower of Hanoi, The eight queens puzzle, etc.).

8. Others Traditional Platforms of E-Learning Used

At the Universidad Nacional de Colombia in this moment is massively used the platform *Blackboard* [36] and recently the virtual learning environment *Moodle* [37] (Figure 11). The learning management system Blackboard usually is used to upload a set of presentations, exercises, homeworks, quizzes, tests on line, links to auxiliary web pages, as well as permits to create forums and another kind of information (Figure 12). The main difference between this class of platform and the platform that we are developing is that our platform permits to create a personal profile automatically as much students as teachers using data mining [38], suggests the form than a student may take the lessons but it is not obligatory since the student may to navigate the content freely and suggests which topics must be reinforce by the student when he/she did not approve the tests.

9. Conclusions and Future Work

The progress made by Universidad Nacional de Colombia (National University of Colombia) in the definition and implementation of an adaptive e-learning system was presented. Content units are modelled in a non-linear structure. This work has shown that it is possible to extend this model to Academic Programs of Universities because courses can be seen as unit contents with relationships between them. It would be possible giving to students the ability to navigate courses of an Academic Program in an autonomous way.

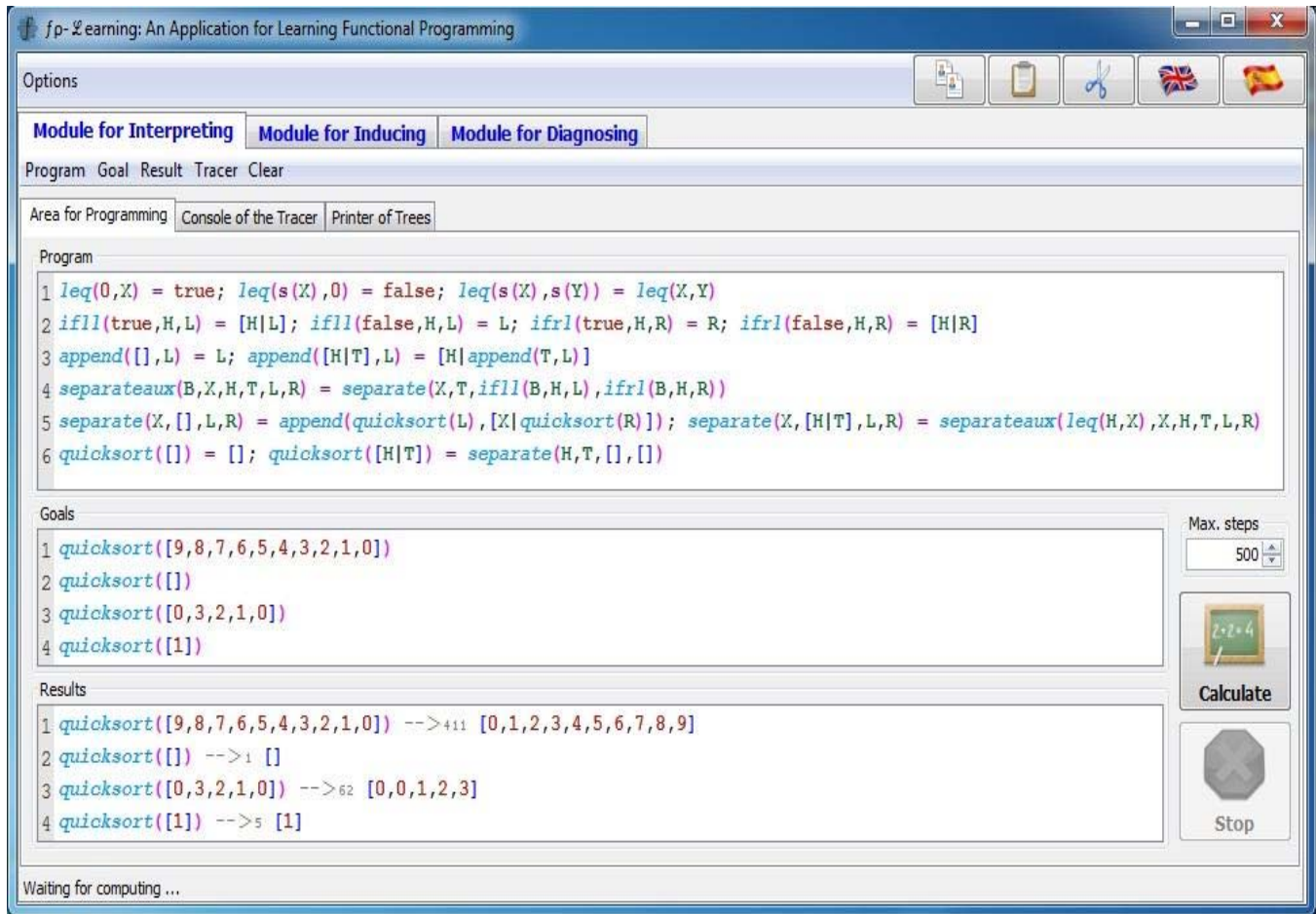


Figure 10. Window for programming functional programs



Figure 11. Others e-learning platforms to teach at the Universidad Nacional de Colombia

It would be interesting to assess the effects of quizzes and short videos in the development of content units. Further research might explore the perception of the student about the platform. An implication of adaptive evaluation is the possibility to motivate students by giving different evaluations in the same way that students are improving their performance.

A definite need for internationalization is observed in the development of the platform to give access to different population groups.

Data Model of this paper has a number of important implications for future practice. Further research might explore social patterns of behavior and courses statistics in order to adapt the platform based on all information about exercises and tests and student progress stored on the platform and to give suggestions to students about their performance.

A basic programming course is presented. An important practical implication is that teaching programming is a hard task. By this reason tools for the learning of functional programming like *fp-Learning* were developed.

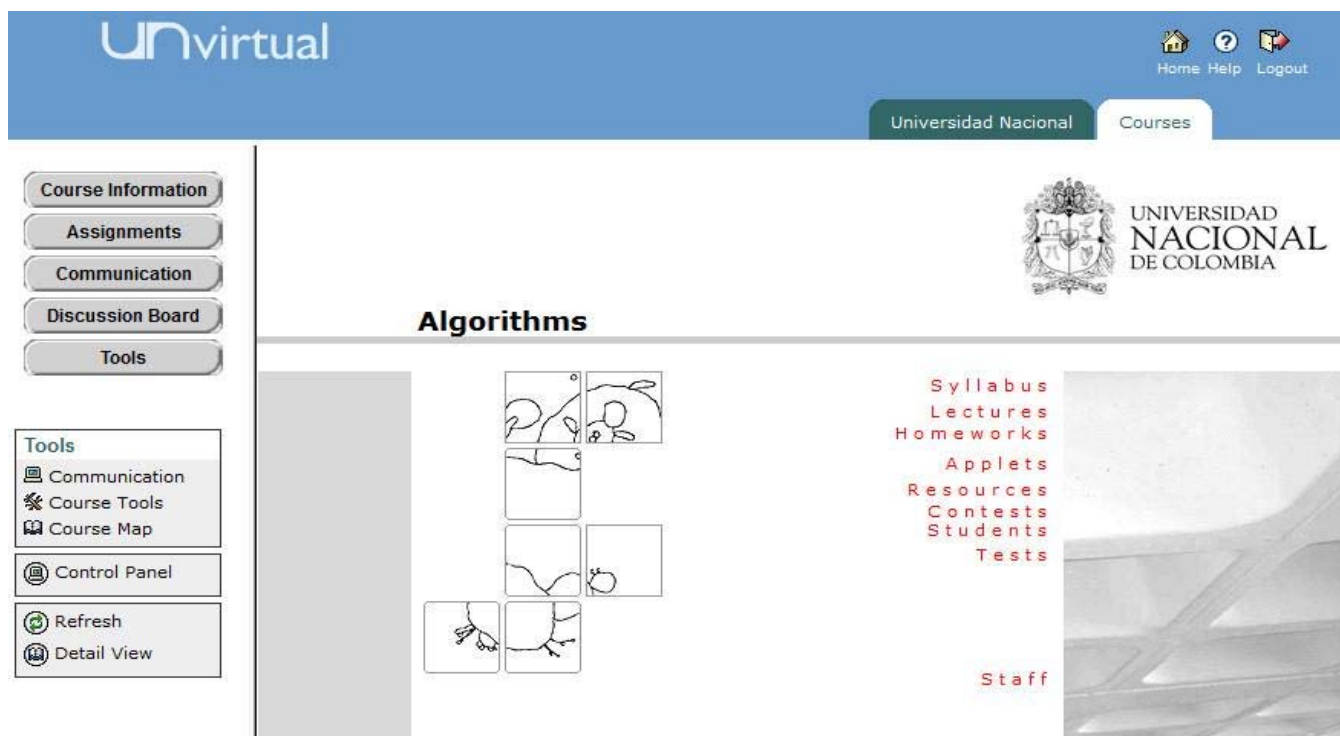


Figure 12. Example of the aspect of the developer of education software Blackboard

References

- [1] Garrison, D. R., Anderson, T. (2003). E-learning in the 21st century: a framework for research and practice. Taylor & Francis.
- [2] Salmon, G. (2004). E-activities: The Key to Teaching and Learning Online, 2nd ed. London: Taylor & Francis, 7.
- [3] —, (2003). E-moderating: The Key to Teaching and Learning Online, 2nd ed. London: Taylor & Francis, 4.
- [4] Kearsley, G. (2000). Online education: Learning and teaching in Cyberspace. Belmont, CA: Wadsworth Publishing Company.
- [5] Bates, A. W. (2005). Technology, e-learning and distance education, 2nd ed. Routledge.
- [6] Zhao, X. (2010). Adaptive content delivery based on contextual and situational model, Ph.D. Thesis, The University of Electro-Communications, Tokyo, Japan.
- [7] Bersin, J., Howard, C., O'Leonard, K., Mallon, D. (2009). Learning Management Systems. Bersin & Associates.
- [8] Balki, R. S. (2010). Learning Management System. Lap Lambert Academic Publishing.
- [9] Ellis, R. K. (2009). Field Guide to Learning Management Systems. ASTD Learning Circuits.
- [10] Gupta, A. (2001). Content management system. Universidad de California.
- [11] Arthur, M. (2006). Expanding a digital content management system: for the growing digital media enterprise. Elsevier.
- [12] Wikipedia, List of content management systems — wikipedia, the free encyclopedia, (2012), [Online; accessed 5-April-2012]. [Online]. Available: [nurlfhttp://en.wikipedia.org/w/index.php?title= List of content management systems&oldid=485497697g](http://en.wikipedia.org/w/index.php?title=List_of_content_management_systems&oldid=485497697g)
- [13] Roger, N. (2010). Advances in Intelligent Tutoring Systems. Springer.
- [14] Murray, T. (1999). Authoring intelligent tutoring systems: An analysis of the state of the art, *International Journal of Artificial Intelligence in Education*, 10, 98–129.
- [15] Looi Chee-Kit, G., MacCalla, Bredeweg, B., Breuker, J. (2005). Artificial intelligence in education: supporting learning through intelligent and socially informed technology. IOS Press.

- [16] Tonidandel, S. (2001). Computer adaptive testing: The impact of test characteristics on perceived performance and test takers' reactions, Ph.D. thesis, Rice University.
- [17] Computerized adaptive testing: theory and practice. Kluwer Academic Publisher, (2000).
- [18] Wainer, H., Dorans, N. J. (2000). Computerized adaptive testing: a primer. Lawrence Erlbaum Associates.
- [19] Garcia-Barrios, V. M., Modritscher, F., Gutl, C. (2005). Personalization versus adaptation? a user-centred model approach and its application, *In: I-KNOW'05*, p. 120–127.
- [20] Stanford University, *Introduction to artificial intelligence*, (2011), last checked: April-4-2012. [Online]. Available: <https://www.ai-class.com/>
- [21] The PHP Documentation Group, Gettext php module, (2012), last checked: April-4-2012. [Online]. Available: <http://php.net/manual/en/book.gettext.php>
- [22] New Digital Group, Smarty template engine, (2012), last checked: April-4-2012. [Online]. Available: <http://www.smarty.net/>
- [23] Slavik, V. (2005). Poedit gettext catalog editor, last checked: April-4-2012. [Online]. Available: <http://www.poedit.net/index.php>
- [24] Baader, F., Nipkow, T. (1998). Term rewriting and all that. New York, NY, USA: Cambridge University Press.
- [25] Ferri, C., Hernandez, O. J., Ramirez, M. J. (2001). Incremental Learning of Functional Logic Programs, *In: the Fifth Int. Symposium on Functional and Logic Programming*, p. 231–247.
- [26] —, (2000). Aprendizaje automático de programas lógico-funcionales, *Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial*, (11) 27–38. [Online]. Available: <http://aepia.dsic.upv.es/>
- [27] Hernandez, O. J., Ramirez, M. J. (1998). Inverse Narrowing for the Induction of Functional Logic Programs, *In: Joint Conference, on Declarative Programming, APPIA-GULP-PRODE'98*, p. 379–393.
- [28] —, (1999). A Strong Complete Schema for Inductive Functional Logic Programming, *In: the Ninth International Workshop on Inductive Logic Programming, ILP'99, LNAI in 1634*, p. 116–127.
- [29] —, (2002). Inductive Inference of Functional Logic Programs by Inverse Narrowing, *Universidad Politécnica de Valencia, Espana*.
- [30] Alpuente, M., Ballis, D., Correa, F., Falaschi, M. (2010). An Integrated Framework for the Diagnosis and Correction of Rule-Based Programs, *Theoretical Computer Science*, 411 (47) 4055–4101.
- [31] Alpuente, M., Correa, F. J. (2003). Un depurador abstracto, inductivo y parametrico para programas multiparadigma, *Universidad Politecnica de Valencia, Tech. Rep*.
- [32] Correa, F. J. (2002). Depuracion declarativa de programas logico funcionales, Ph.D. Thesis, Informatica, Universidad Politecnica de Valencia, Valencia, Espana.
- [33] Comini, M., Levi, G., Meo, M. C., Vitiello, G. (1999). Abstract Diagnosis, *Journal of Logic Programming*, 39 (1-3) 43–93.
- [34] Comini, M., Levi, G., Vitiello, G. (1995). Declarative Diagnosis Revisited, *In: Proceedings of the 1995 Int'l Symposium on Logic Programming*, Lloyd, J. W., Ed. The MIT Press, p. 275–287.
- [35] Shaphiro, E. Y. (1982). Algorithmic Program Debugging. Cambridge, Massachusetts: The MIT Press, ACM Distinguished Dissertation.
- [36] Wikipedia, Blackboard inc. — wikipedia, the free encyclopedia, (2012), [Online; accessed 1-June-2012]. [Online]. Available: [http://en.wikipedia.org/w/index.php?title=Blackboard Inc.&oldid=495683476](http://en.wikipedia.org/w/index.php?title=Blackboard+Inc.&oldid=495683476)
- [37] —, (2012). Moodle — wikipedia, the free encyclopedia, [Online; accessed 8-June-2012]. [Online]. Available: <http://en.wikipedia.org/w/index.php?title=Moodle&oldid=496174427>
- [38] Shahabi, C., Zarkesh, A. M., Adibi, J., Shah, V. (1997). Knowledge discovery from users web-page navigation, *In: Proceedings of workshop on research issues in Data engineering*, p. 20–29.