Web based education and assessment in the Bologna process

Ezequiel López-Rubio, Enrique Domínguez-Merino, Gonzalo Ramos-Jiménez

Department of Computer Languages and Computer Science
University of Málaga
Bulevar Louis Pasteur, 35
29071 Málaga
SPAIN

{ezeqlr,enriqued,ramos}@lcc.uma.es

ABSTRACT

The creation of the European higher education area by means of the Bologna process implies a series of changes in teaching and evaluation methodologies. These include emphasizing the participation of the student and the incorporation of new technologies such as web based education. Hence, a full restructuring must be carried out for most subjects, which must take into account new tradeoffs between the contents and the aims of the students. Moreover, teaching strategies and practices must be updated accordingly. Here we discuss a case study in full, which is based on our own experience. In order to adapt the Automata Theory subject to the new structure of Computer Science degrees, a novel teaching methodology has been developed that belongs to the ECTS credit system implantation experience sponsored by the Computer Engineering School of the University of Málaga. This methodology emphasizes assessable classroom activities, along with different kinds of projects to be completed during the course. A wide range of optional assignments is provided in order to cover the variability of students' interests. Furthermore, the performance of the students is evaluated through the course with the help of web based assessment tools, so that the student is always aware of his/her progresses. These tools are centralized in a virtual campus, which also serves as a framework for additional activities such as discussion forums and collaborative tasks. These adaptations are supported by a careful design of the way that the theoretical concepts are introduced. In particular, the notation, the presentation style and the relative emphasis on the different sections of the syllabus are guided by students' capabilities and the overall goals of the degree. These modifications should result in an improvement of the motivation and the integration of the contents of the subject with the Computer Science body of knowledge that the School tries to convey.

Keywords

Web based education, web based assessment, Bologna process, virtual teaching.

1. Introduction and objectives

Automata Theory sets up the bases for the construction of physical devices which are capable of carrying out automated computations. By means of a mathematical framework that makes use of tools coming from Algebra, Logic and Discrete Mathematics, it models different kinds of devices which are key parts of computer systems, while at the same time it proves some of the properties of these models, and determines their limitations. Because of these reasons, we believe that it is a fundamental subject in the basic formation of computer scientists.

In order to adapt the subject to the new structure of Computer Science degrees, which follows the Bologna process, we have developed a teaching methodology that belongs to the ECTS credit system implantation experience carried out by the Computer Engineering School of the University of Málaga. This methodology emphasizes the assessable classroom activities, along with different kinds of projects to be completed during the course. A problem arises when it comes to evaluate the performance of the students, since there are many marks which must be combined in order to obtain the final mark. The tools provided by the online assessment system of the University of Málaga (named AlfilWeb) and the virtual campus (based on Moodle [1,2]) were not adequate for this task, due to the simplicity and rigidity of those systems (see Figure 1). Nevertheless, a new version of the virtual campus has included new features, in particular a grade computation module which is particularly useful for our purposes. It does not provide full spreadsheet capabilities, but rather it allows a reduced set of functions. Hence, we have had to express the rather complex evaluation rules of the Automata Theory subject by means of that reduced set of functions. The AlfilWeb system remains useless, since it only computed weighted averages, with a maximum of 8 partial marks. On the other hand, the virtual campus allows an unlimited number of activities with different assessment systems (see Figure 2), which is an advantage of Moodle over other course management systems [3,4,5]. Before the update of the virtual campus, the students were not able to see their marks in Moodle, and they did not have a clear idea of their evaluation during the course, since they had to wait for the teachers to compute the marks manually.

Because of this situation we decided to design a tool which takes advantage of the virtual campus features, in order to facilitate the activity evaluation task for this subject while at the same time it offers the students all the available information about their progress in a user friendly way. These tools were intended to reduce the time invested by the teacher in the assessment of the activities, so that the saved time could be redirected to more significant educational tasks, thus improving the quality of the education. Moreover, the motivation of the student during the course was expected to increase, since he/she would be informed of his/her progression. All these goals will be of great help in order to implement the new structure of the BSc degrees in Computer Science, which includes the subject (with the above described methodology) as compulsory in the Sophomore (second) year.

Hence we petitioned and obtained funding from the University for a Pedagogical Innovation Project which tries to fulfill these needs. It has been carried out during the academic years 2008-2009 and 2009-2010. The objectives of the project are the following:

–Improve the assessment and tracking system of the performance of the students in the subject, in order to ease the work of the teachers and improve the motivation of the students.

–Verify the viability of the newly designed methodology in order to prepare its implantation in the new structure of the degree.

The structure of this paper is as follows. First we explain the academic context of the Automata Theory course (Section 2). Then we outline how the new course has been designed to incorporate a web learning approach (Section 3). After that we explain what has been done in the web assessment project (Section 4). Finally, we summarize the results of the project (Section 5) and draw some conclusions (Section 6).

2. Academic context

The way to make the transition to the ECTS credit transfer system fueled an intense debate in Spain. At last, the Government decided to set a common structure for nearly all BSc degrees (including all those of engineering), so that they will have 240 credits (4 years). In some particular cases (like Medicine) there will be longer degrees. After that you might pursue a MSc degree, usually with 60 credits (1 year).

But perhaps the most important change is in the methodology. Special emphasis is to be given to practical learning. Put in simple words, this implies fewer theoretical lectures and more laboratory sessions. At the same time, the final exam will have its weight reduced.

In which respects to Computer Science (CS) degrees, a common framework has been designed for them in Spain, following the recommendations of ACM/IEEE CS curricula. Five CS undergraduate curricula are defined:

- Computer Engineering (hardware).
- Software Engineering.
- Computation (fundamentals).
- Information Technology.
- Information Systems.

In particular, at the Computer Science School of the University of Málaga we have implemented this scheme by means of three BSc degrees:

- BSc in Computer Engineering.
- BSc in Software Engineering.
- BSc in Computer Science, with 3 branches (Computation, Information Technology and Information Systems).

The first two years are common to all degrees, so that the students can switch from one degree to another seamlessly until they finish their Sophomore year. Automata Theory is a core course in the first semester of the second year.

This implies that the course will be taken by CS students with very different aims. For example, for Information Technology students this could be the last course with a strong mathematical content. In other cases, such as the Computation branch, this will be an initiation course for further study in formal methods and even an introduction to research.

For these reasons, we are strongly committed to improve the students' motivation, which tends to be low in theoretical courses. Our way to achieve this goal is to emphasize web based learning, as we see in the next section.

3. Web learning

Next we outline the design goals of the course. The material to be covered is divided into two sections:

- Formal languages: grammars, finite automata, pushdown automata.
- Computability theory: Turing machines, Church-Turing thesis, halting problem, Rice Theorem.

Our students have a hard time trying to understand complex notations, and most of them are not keen to follow mathematics courses. To make things worse, there are several world-wide used books [6,7] each with its own notation. In order to simplify matters as much as possible, we chose to use only the notation in [6]. Moreover, we made some adjusts in the definitions, such as removing rules with empty rhs from most types of grammars so that most algorithms get much easier to understand. Finally, Turing machines are not used for the main results in computability theory because of the inherent difficulty of large machines; WHILE programs are used instead, which appeal to the programming experiences of the students.

We try to make it worthwhile to pay a continuous attention to the course during the semester. This is because theoretical courses are largely ignored by the students until the last days before the final exams. We avoid this tendency by setting out a series of activities along the course which contribute to the final grade. Also, we emphasize the connections with other courses: Algebra, Algorithm Analysis, Compilers, etc.

With all of this in mind, we have designed several tools and activities. One of the most popular is the algorithm training software (see Figure 3). It is a Java program which implements all the algorithms of the course with our notation. Furthermore, the execution can be traced step by step, and you can create your own inputs and supply them to the tool. Students have found it to be very useful to understand the algorithms and prepare the final exam.

Other web activities, all of which are based on the virtual campus, are the following: online tests, tasks, workshops, forums and wikis. The online tests provide the student with his/her mark as soon as the test is finished.

Finally, we also have other activities that are not so interactive. These include:

- An online repository of past exams (with solutions).
- A poster creation task (to be stuck to the walls of the classroom).
- The preparation of class presentations on topics related to the course: mind-machine, history of algorithms,…
- A final lecture on future perspectives: artificial life, quantum computing,…

The most recent improvement is the assessment tool, which is integrated into the virtual campus. Its features are:

- Complex grading formulas can be used.
- Every student is able to see his/her marks in real time.
- This way the student is able to track his/her achievements as the semester goes.
- Students’ motivation is greatly improved.

The construction of the assessment tool was the objective of the web assessment project that we describe in the next section.

4. Web assessment project

The project was split into the following phases:

1. A preliminary study of the requisites of the tool to be developed. To this end we carried out surveys among the students to know their preferences, and the course instructors were consulted. The data acquisition by means of the student surveys and the interviews with the instructors was done by the teachers who participated in the project.

2. Software development. We started with a preliminary design, and then we implemented the final version of the tool, which was integrated in the virtual campus.

3. Software tests and evaluation. The tool was tested and evaluated by the students and the instructors. The observed bugs were corrected.

4. Software utilization. The tool has been used during the academic year 2009-2010, once we got a tested and correct version. It has been used routinely in the course for the evaluation of the class activities and the remaining tasks done by the students.

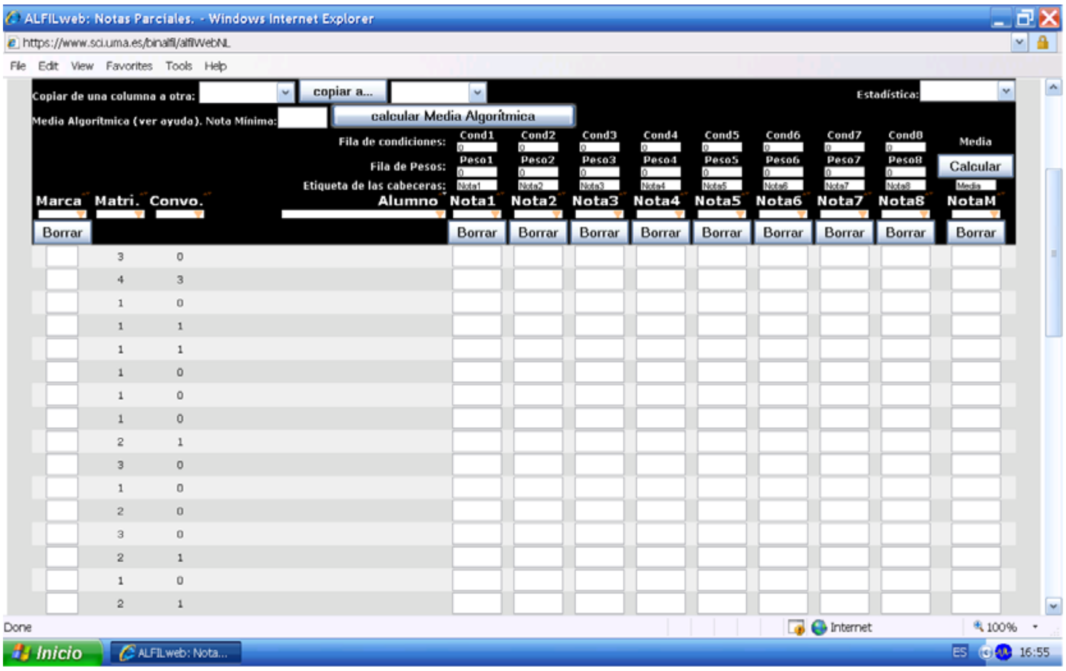

Figure 1. Mark computation page of the AlfilWeb student tracking system. Please note that the students' names have been hidden.

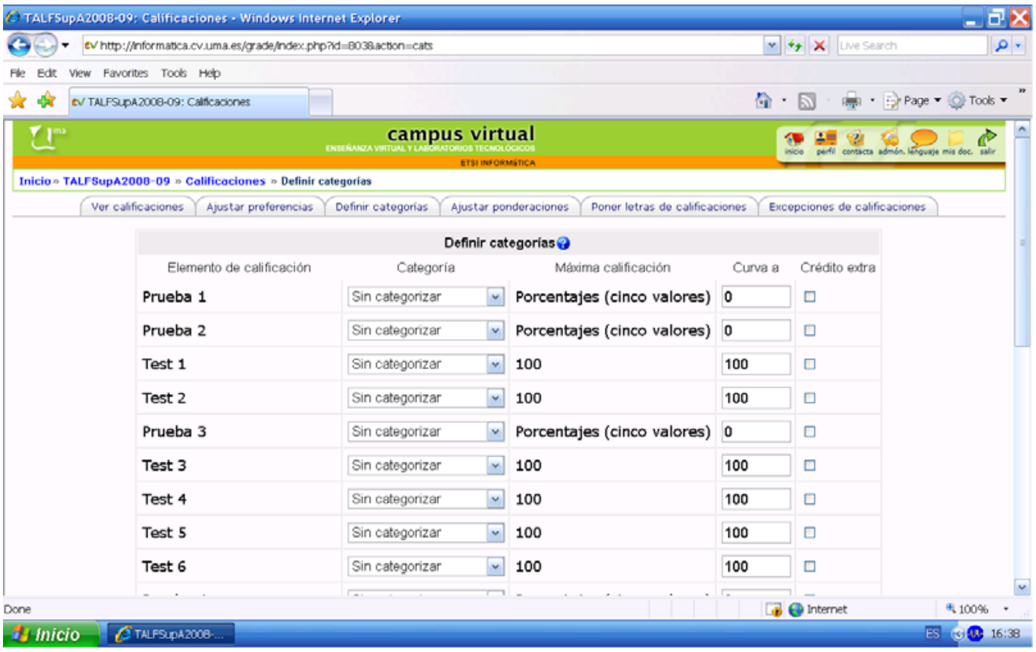

Figure 2. Grading categories definition web page (virtual campus).

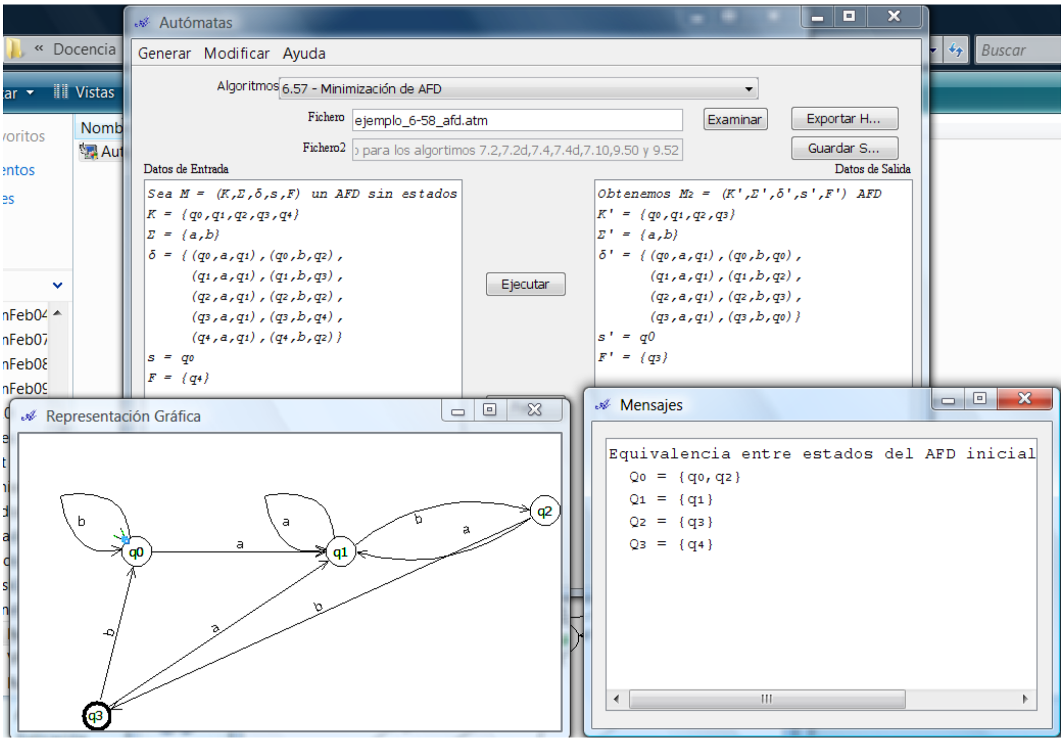

Figure 3. Algorithm training software (minimizing a Deterministic Finite Automaton).

We discovered the need of support personnel to help to implement the proposed tool. We selected a student who had enrolled the course some years ago. The teachers guided him in every moment in order to comply with the proposed goals.

The project was evaluated by students and teachers in the Phase 3 above explained, so as to find out whether the software suited the users' needs, and then it was evaluated again in the Phase 4, when it was being used routinely in the course for the evaluation of class activities and other tasks. The students evaluated the project by means of surveys, and the instructors by means of interviews.

5. Project results

During the academic year 2008-2009 we carried out the following activities:

a) Survey design. We made a questionnaire which was aimed to find out the preferences and attitudes of the students with respect to the integration of the course in the virtual campus. We made every effort to pay more attention to those aspects which could give us a clearer idea of what the students need to improve their motivation when it comes to study Automata Theory.

b) Survey implementation. The instructors have done the survey, and they have tried to obtain well reasoned answers so as to increase the validity of the results.

c) Analysis and interpretation of the obtained data. We have processed the obtained data, which yield a favourable opinion of the students to the purpose of the project.

d) Elaboration of the specifications of the software. From the results of the survey and the teachers' previous knowledge and ideas, we have obtained a specification of what the software tool for the assessment and tracking of the students should do.

e) Support personnel selection. We searched for support personnel to help with the software development. We found an old student whose programming knowledge was very useful for our purposes. We met with him to explain and discuss the programming aspects of the project.

f) Preliminary software design. We got a preliminary prototype which was able to evaluate some of the necessary mark calculations, although it did not have all the desired capabilities.

During this academic year 2009-2010 we completed the remaining tasks:

a) Final software design. We developed a final version of the software tool which complied with all the relevant specifications.

b) Preliminary tests. Once the software was developed, the teachers tested it before using with the real marks of the course. This was done to avoid possible errors which could lead to data loss or other undesirable consequences.

c) Final tests. After we checked that there were no serious failures, we started to use the software to convey the marks to the students, and they gave us their opinion about its performance, and also about possible improvements. All of this has been done while the newly designed methodology for the subject was being implemented.

d) Software enhancement. From the students' suggestions and the spotted bugs, we did a new cycle in the software development process, so that its performance was increased.

e) Evaluation of the results. We carried out a final evaluation of the tool, in order to determine whether the objectives of the project were completed. As it was found that it was the case, we further studied the necessary adaptations when the new degrees adapted to the ECTS credit system are established.

Despite the good results obtained in the activities of the first year, the final design of the software had a slight delay, due to the fact that the old student in charge of the software development was working for us only part time. However, this did not compromise the project, since we had enough extra time to compensate for this delay.

6. Conclusions

We have adapted the Automata Theory course to the needs of the new ECTS credit system. The transition to the ECTS implies a major effort to redesign all the courses and degrees. Automata Theory has been adapted to address the variability in students' goals. The new focus on web based learning improves students' motivation and achievements.

As a final preparation for the transition, a web assessment tool was developed by means of a project. The general outcome of the project has been satisfactory, since the academic year 2008-2009 yielded a clear idea of the needs which the software needed to fulfil, from the results of the surveys and the interviews with the teachers. This facilitated the subsequent phases. Moreover, we found support personnel who completed some of the necessary tasks. Finally, in the academic year 2009-2010 the final version of the software was deployed, and the students and teachers found that it was adequate for the assessment of the Automata Theory course.

As a final remark, we must say that there is a need to ensure that the new methodology does not prevent in-depth understanding of the fundamental concepts.

References

[1] Moodle web page: http://www.moodle.org

[2] Cole, J., (2005). Using moodle: Teaching with the popular open source course management system. Sebastopol, CA : O'Reilly Community Press.

[3] Beatty, B. & Ulasewicz, C. (2006). Faculty perspectives on moving from blackboard to the moodle learning management system. Tech Trends: Linking Research & Practice to Improve Learning, 50(4), 36-45.

[4] Bremer, D., & Bryant, R. (2005). A comparison of two learning management systems: moodle vs blackboard. Proceedings of the 18th Annual Conference of the National Advisory Committee on Computing Qualifications, Tauranga, New Zealand.

[5] Corich, S. (2005). Let's get ready to moodle. Bulletin of Applied Computing and Information Technology, 3(3), 1-7.

[6] Lewis, H. & Papadimitriou, C. H. (1997). Elements of the Theory of Computation, 2nd Edition. Prentice-Hall.

[7] Hopcroft , J. E., Motwani, R., & Ullman, J. D. (2006). Introduction to Automata Theory, Languages, and Computation, 3rd Edition.