Sánchez, Carles; Ramos, Oriol; Márquez, Patricia; Martí, Enric; Rocarias, Jaume; Gil, Debora

AUTOMATIC EVALUATION OF PRACTICES IN MOODLE FOR SELF LEARNING IN ENGINEERING

AUTOMATIC EVALUATION OF PRACTICES IN MOODLE FOR SELF LEARNING IN ENGINEERING

Carles Sánchez[1], Oriol Ramos[1], Patricia Márquez[1], Enric Martí[1], Jaume Rocarias[2], Debora Gil[1,3]

[1]Comp. Vision Center, Comp. Science Dep. UAB

[2]Information System Planification UAB

[3]Serra Hunter Fellow

Spain

csanchez@cvc.uab.cat, oriolrt@cvc.uab.cat, pmarquez@cvc.uab.cat, enric@cvc.uab.cat, jaume.rocarias@uab.cat, debora@cvc.uab.cat

Abstract

The first years in engineering degree courses are usually made of large groups with a low teacher-student ratio. Overcrowding in classrooms hinders continuous assessment much needed to promote independent learning. Therefore, there is a need to apply some kind of automatic evaluation to facilitate the correction of exercises outside the classroom. We introduce here a first experience using surveys in Moodle 2.0 in order to get an automatic evaluation of practices in our Database course. We report survey valuation of the autonomous learning tool and preliminary statistics assessing correlation to an improvement in the practice exam marks.

Keywords – Large groups' evaluation, Autonomous learning, Engineering studies, Database practices, Moodle surveys.

----

1 INTRODUCTION

Database course is a core compulsory subject in year 2 in Computer Engineering at the Universidad Autónoma de Barcelona (UAB) from 6 ECTS. The course introduces basic contents Database (DBMS) management systems and contents are divided into two blocks: DBMS relational design and manipulation of DBMS relational using Structure Query Language, SQL (Date 1987). One of the core skills is the ability of holding consultation of some complexity using SQL in client-server architecture, therefore practical exercises totalling 2.52 ECTS (41%) are distributed in 0.48 ECTS practice sessions and 2.02 autonomous individual exercises.

In previous years, students had access to an account on the database server, the DB with which they would work throughout the course, and a set of queries in SQL language to do together with the results. In consultation class time was spent in group discussion in order to answer any doubts that may arise. This was considered independent student work and an assessment of it was done during a practical test.

The emergence of new technologies in evaluation platforms together with the success of the first experiences in their application (Arenas, Molleda, Chávez, Domingo & Castañeda, 2003) arouse our interest in using them for SQL language learning in Database courses. Besides these kind of technological tools can provide formative assessment as established by the European Higher Education Area (EHEA). Formative assessments are evaluation systems that allow the gathering and interpretation of data about the student progress in order to provide him/her with a feedback on his progress (Rotger, 1990). Recent studies (Zaragoza, Luis Pascual & Manrique, 2009) have shown these systems assessment possessed considerable advantages for optimizing the learning of students: greater involvement and motivation of the students, performance improvement, both in relation to learning as regards the final rating.

Journal of Technology and Science Education. Vol 5(2), 2015, pp 97

On-line ISSN 2013-6374 – Print-ISSN 2014-5349 DL: B-2000-2012 – http://dx.doi.org/10.3926/jotse.146

Under these considerations, we decided to use our materials of SQL queries with solutions and code available for several already created databases to design and implement tools for their immediate evaluation. We also decided to implement these evaluation tools on Cerbero (Cerbero, 2014) which is the Moodle 2.4 platform we use in our course to manage on-line interaction with students.

Our experience in the development of self-learning and automatic evaluation tools started the academic year 2012-13, when we considered the possibility of automating the correction of practical tests with the dual purpose of

- ease the load of work of teacher's correctors (Martínez & Crespo, 2007) and

- give the students a faster feedback on their results.

To this end, students were asked to send their results using a SQL code in a very concrete form through the Moodle program used in the course; this was then automatically checked by a very simple home-made php script that evaluated the SQL code on our server.

Encouraged by the positive results of the aforementioned experience, we decided to implement autonomous assessment methodologies in the academic year 2013-14 that would provide students with an immediate evaluation so they can assess their own degree of maturity in performing SQL queries. The objectives of this on-line tool for systematic evaluation of SQL queries are:

- Students' self-assessment of the SQL queries that provide and immediate result of the year's corrections.

- An efficient assessment of the practical problems by teachers to provide students with an approximate grade on test day.

As a side result, this tool also allows efficient and quick collection of learning's evidence that allow significantly improvement in individual continuous assessment of students in large groups.

2 WORK DESCRIPTION

Manipulation (queries) over a Data Base (DB) using SQL constitutes 41% of ECTS of the subject, distributed in 6 tutored practice sessions of 2 hours and 51 hours of individual practice without a teacher. In the year 2013-14 the number of registered students was 260 and they were distributed in 10 groups of 28 students each for practical sessions. Concerning teaching staff, there was a teachers' team of 2 doing theory/problems and 3 teachers supervising the practical classes. This teaching load made the monitoring of a periodical assessment delivery unfeasible, as the teachers' feedback cannot be as fast as required. This situation encouraged the development of alternative evaluation methodologies providing an immediate feedback to students.

Since 2011, we have used a Moodle platform for the publication of materials, deliverables and surveys; last year that was upgraded to Moodle 2.0 (Cerbero, 2014). Moodle (Cole & Foster, 2007) is used in a variety of institutions and agencies and the new upgrade includes tools for group work (group formation, deliveries, surveys, self-assessment and control) that have been used successfully in other subjects (Martí, Rocarias, Gil, Vivet & Julià, 2008). We have created a new module called SQL Queries (SQLQ) based on the survey module already available in Moddle 2.0. This modification of the Moddle survey module incorporates new options that are available depending of the user's profile:

Teacher user: When creating a new activity SQLQ the form allows the teacher to introduce a set of SQL queries with their solutions SQL and the due dates for delivery. The output of the teacher SQL query is compared to the output of the student code to validate his/her solution. The comparison is performed by checking that the contents of tables obtained from executing the teacher and student SQL codes have the same values for each row and column.

The activity also has an option to set the number of times that a student can validate his/her SQL code for a given query. In the case of repeated attempts we have a self-assessment system, however, if the students find only one possible try the activity becomes a test of automatic correction. Furthermore, being an activity derived from the activity survey's module, it can be configured so that the teacher can specify the range of IPs accepted. Figure 1 shows the teacher environment for the definition of a new activity SQLQ in case of an examination. First, the name of the activity and its description must be introduced. Then, set the permitted IPs to use the activity and the test modality by selecting the number of attempts. Finally, introduce the test solutions.

Figure 1. Teacher environment for the definition of an SQLQ activity in case of examination

For each activity SQLQ in use, the Moodle module provides the teacher with several tools to analyze the student's responses to each query and allows downloading the results in an Excel file. The module offers several statistics of the rate of student's success and the possibility to compare the student code with the teacher solution. The statistical summaries are computed for each student, each query and also for all students and queries to provide with a global indicator of the rate of student's success. For each statistical summary, the teacher environment shows the percentage of correct and wrong answers and the percentage of queries that have not been answered in case of an examination. Figure 2 shows the environment presenting the statistical summary in case of a query based analysis. On top of the screen there is a bar menu that allows switching to one of the available summaries and below there is the table reporting statistics. The module also provides the comparison between student and teacher SQL solutions. This facility is especially useful in case that the student requests a revision of his/her results. Figure 3 shows the environment to compare codes for a given student in a 12-query test activity. The most left column shows the mark obtained by direct comparison of the output obtained by the two SQL codes: 2.0 if student and teacher tables are equal, 0.0 otherwise.
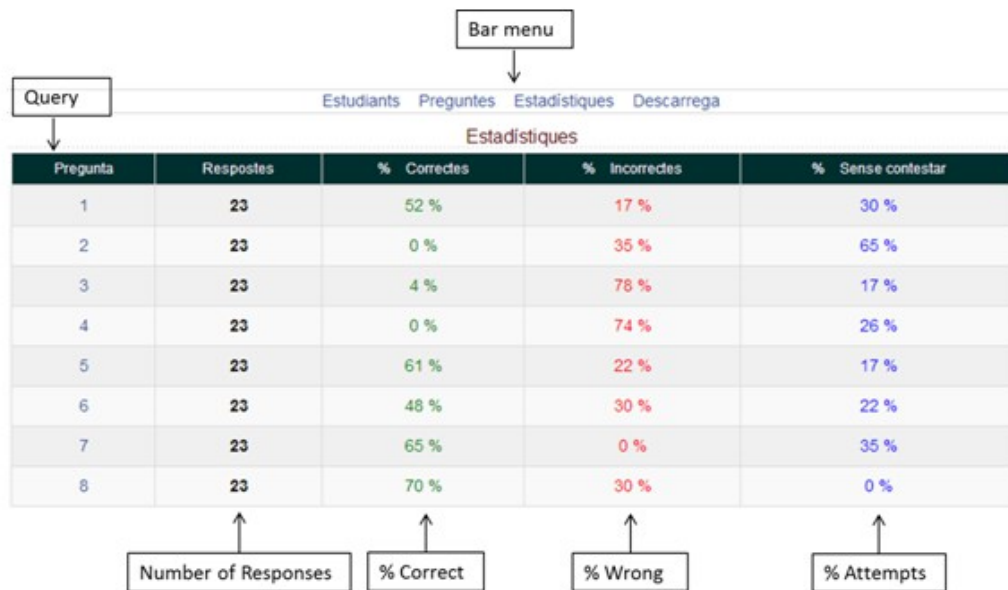
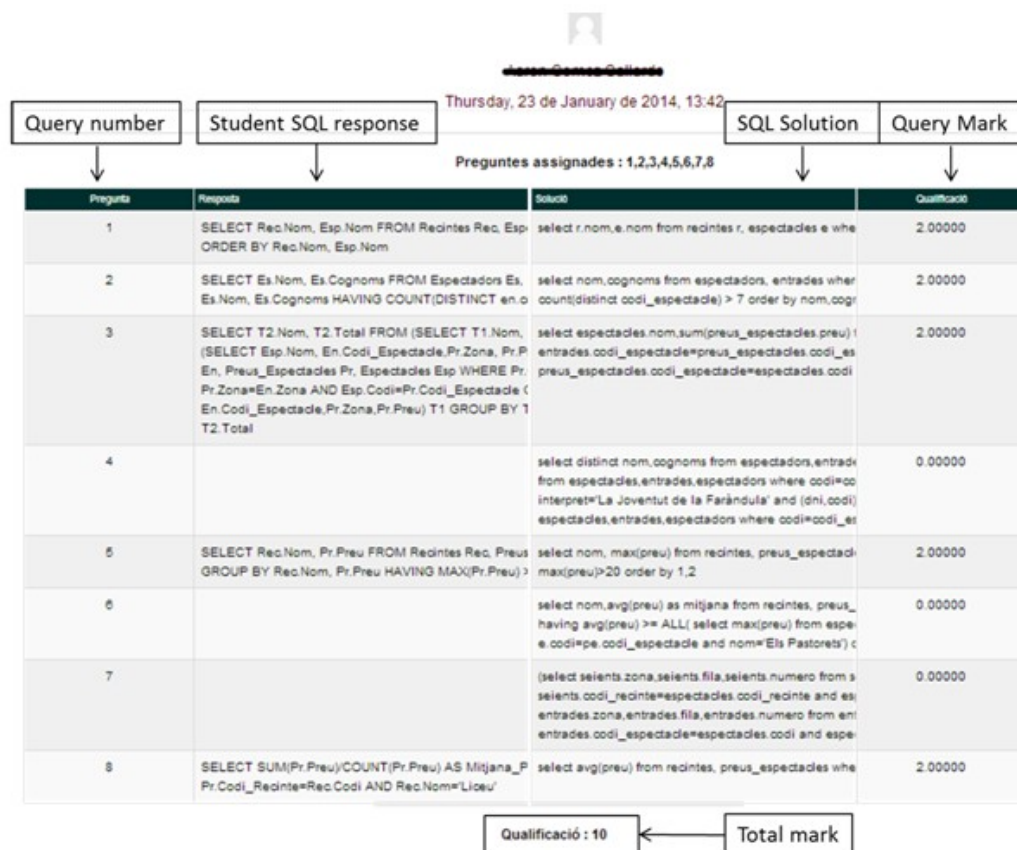*Figure 2. Teacher environment to analyse student queries: general statistics*

Figure 3. Teacher environment to analyse student queries: comparison between teacher and student SQL codes

Student user: When a SQLQ activity is created the student can access it to send SQL queries that he/she has previously tried on his/her own. Figure 4 shows the student environment to introduce his/her solutions to the test queries. On top, due dates are displayed and the bottom dialog box is the space where the student can write his/her SQL solution to one of the queries (indicated by numbers) from a list that has been previously distributed to students.

Figure 4. Student environment to introduce his/her SQL code solving the queries proposed in the test

In case of self-assessment students can try the queries as many times as he/she wants and in any order while the activity remains open. By submitting the form SQLQ the system returns the number of hits and the students have detailed access to their responses to check which queries have been correctly answered. In the case of an examination, the student has only a single attempt, and then he/she receives his/her marks immediately. Figure 5 shows the student environment providing feedback for each of the queries he/she has made. The total grade is displayed on the bottom of the screen.
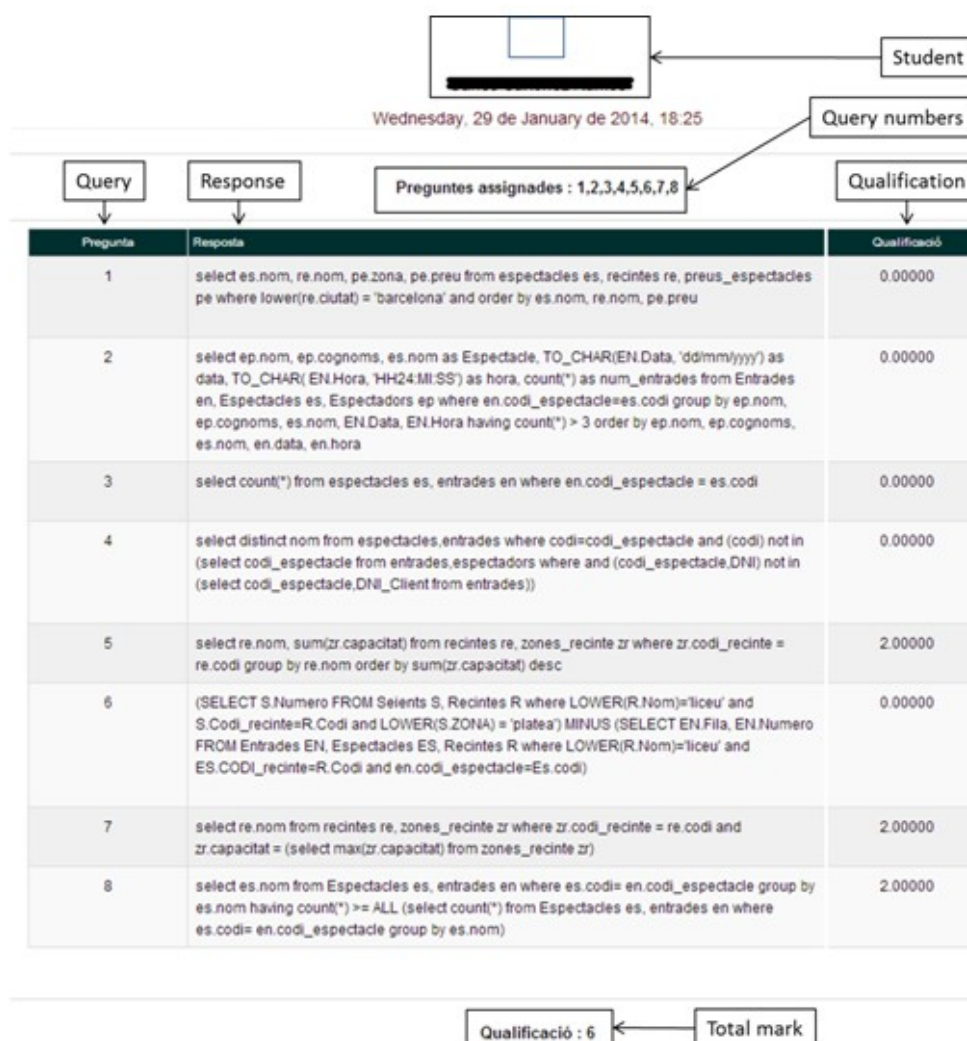
Figure 5. Student environment to analyze the queries made

3 RESULTS

The overall assessment by teachers is positive, given the improvement in the grades that a first statistical comparison indicates. Figure 6 (left) shows a bar chart comparing the practices' grades of the year's 2012-13 students (without self-evaluation) with the current course. The bar chart shows that in the current year there is a lower percentage of students having marks in the range 0-4, and that the percentage of students in the range of 10 is much higher than in the previous year. In conclusion, more students had passed their exams this year and there is also an increase in the percentage of those with higher grades. The average mark for the year's 2012-13 students was 6.0 and 7.5 for current course students. Finally, a t-test for unpaired data (Montgomery, Runger & Hubele, 2009) at significance 0.05 detects a significant difference between average marks (p-val = 1.1170e-007) with a confidence interval for the difference in means between the course 2012-13 and the current one equal to [-2.0464, -0.9564].
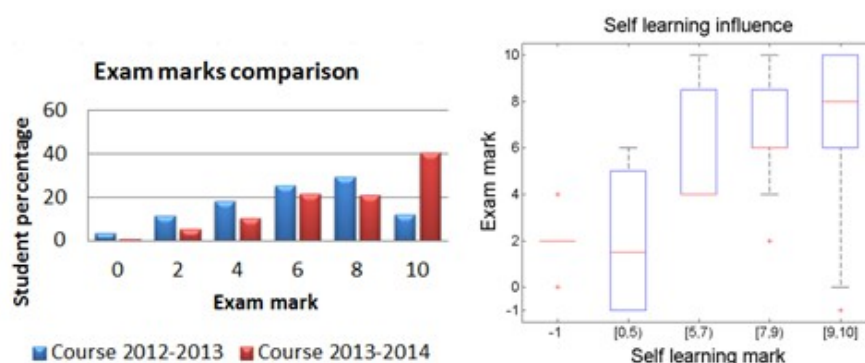
Figure 6. Results of student assessment

To better assess the impact of self-assessment module in final practice, in Figure 6 (right) we compared the module's exam grade with the exams in a flow chart. We have grouped the self-assessment grade: not submitted, -1 suspense, [0,5) pass, [5,7), remarkable, [7,9) and excellent, [9,10]. In the case of the practices' exam, the value -1 in the boxes indicates students who did not attend it. The graph shows a qualitative visual dependency between both grades with an increased tendency in the averaged exam's grades. The number of failed students is the same than those who had failed or got a close pass in the practices' exam. Also, students with good grades in the self-assessment module have obtained the highest grades in the practices' exam, with the exception of the lower percentile of 25% in the excellent group. We attributed these exceptions to three main factors. Possible cheating or copying among students in the self-learning module, errors due to exam jitters or ineffective self-learning tool. However, the total number of these exceptional cases is 20 students, which represents only 7.6% of the total of 262 students enrolled in the course and 7.9% of the 251 submitted. Therefore, given the mean and percentiles group of excellent notes, we can draw two main conclusions. The first conclusion is that the tool has benefited the vast majority of students who have used it. This follows from the fact that most of the best marks obtained in the examination of practices corresponds to those students who have used the self-assessment tool more intensively. The second is that most students have legitimately used this tool and is an indication of the degree of acceptance thereof.

The acceptance and valuation of the module among students has been assessed by an anonymous opinion survey. This survey is a Moddle module that we have implemented using the available Moodle survey module to allow anonymization among other facilities. We report global statistics for the general valuation of the self-assessment tool and the practical classes in the bar plots shown in Figure 7. The results obtained from the survey valuation show that most students value positively the autonomous learning tool with a 7 on average. There is, however, a significant percentage of students that have a neutral valuation (5) of the tool. We attribute this neutral perception to the fact that this is the first year that the activity has been proposed and, consequently, during the first weeks the activity had to work in prove mode and students were requested to report any incidence. Errors reported by students helped in improving the module and develop a stable version that was the one used to do the self-assessment activity. Still the perception of the practical activities was very positive (see bar right plot in Figure 7) with a clear average of 8 points out of 10.
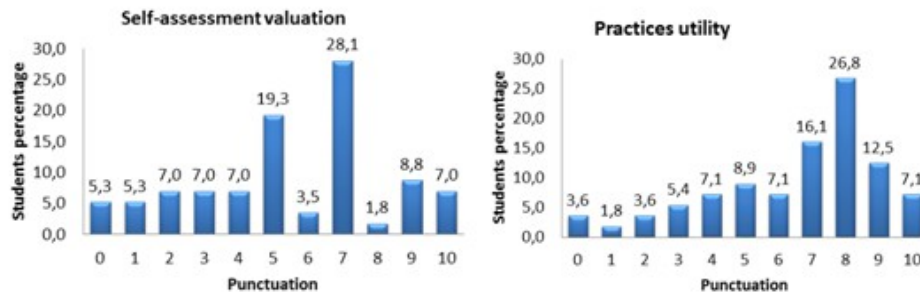
Figure 7. Results from student's survey for validation of self-assessment tool (left)
and practical classes (right). Worse punctuation 0 and best 10

In order to make a more precise analysis of this module on the student's results, the relative weight of the correct number of queries in the self-assessment module in the final grade must be taken into account. As mentioned earlier in this work, in previous years we did not have any tools that would allow us to assess in an individual and continuous way the progress made by the students in mastering the SQL syntax and, therefore, had no way to assess directly the autonomous work done by them. Thanks to this module, while students receive an immediate return on their progress, teachers are able to collect evidence that can be used to mark them. That has allowed us to allocate 10% of the final grade for this item, which accounts for 25% of the 2.52 ETs assigned to practical exercises. We think that this weight has definitely encouraged the students to use the self-assessment module.

4 CONCLUSIONS

We present in this paper a first experience using surveys in Moodle 2.0 in order to get an automatic evaluation of practices in our Database course. A first valuation of this experience allows us to notice, in one hand, a relation between an improvement in the practice exam marks and the use of the self-learning module, and, in the other hand, a decreasing workload for the teachers. Despite all the positive aspects of this first experience we have observed some limitations that could be improved in the future. On the one hand, it has been implemented using the Moodle survey tool, which limits the feedback we can offer to students whether the query is successful or not. With the current module we have to prepare an activity for each group of practices, which share the same array of questions. With the module questionnaire we could have generated a single activity where each student could have had different random questions to answer from a set of questions grouped by difficulty and type. On the other hand, if we had used the questionnaire module we could have used its full potential to (a) provide better returns to students according to the failed response and (b) facilitate the generation of practical exams.

The current SQLQ module is based on Moddle, which only provide a limited feedback to students. After our first positive experience using this module, our teaching team is developing an activity version based on Moodle questionnaires. Being based on questionnaires, this new module will manage activities, groups and feedback given to the student. It will also allow to incorporate generic models of corrections (done in php, for instance) of other types of programming practices, such as C programming or image processing, which verification could be based in the quality of output produced. We will make this new module based on Moodle questionnaires public after fully evaluating its performance as autonomous learning tool and its acceptance among students. This autonomous learning module will allow us to add small progress evaluations at the end of each practice session. This year we have used them to test the module before using it in evaluation activities. Once evaluated, it can be used to rate the progress of students in each practice session.

Finally, although the preparation of the set of solved queries requires some time and effort in order to get a reliable and complete set, the first impression among all teaching staff is that such effort is worth the time saved by professor in comparison to the conventional system requiring manual corrections.

REFERENCES

Arenas, F.J., Molleda, G., Chávez, M.E., Domingo, M.A., & Castañeda, R. (2003). Evaluación del autoaprendizaje con herramientas basadas en las tecnologías de la información y las comunicaciones. *Revista de Enseñanza Universitaria*, 23, 7-22.

Cole, J., & Foster, H. (2007). *Using Moodle: Teaching with the Popular Open Source Course Management System.* O'Reilly Community Press.

Martí, E., Rocarias, J., Gil, D., Vivet, M., & Julià, C. (2008). *Uso de recursos virtuales en Aprendizaje Basado en Proyectos. Una experiencia en la asignatura de Gráficos por Computador*. VIII Jornada sobre Aprendizaje Cooperativo, I Jornada sobre innovación Docente, Lleida, July 2008.

Martínez Lirola, M., & Crespo Fernandez, E. (2007). La evaluación en el marco del EEES: El uso del portfolio en Filología inglesa. *REDU. Revista de Docencia Universitaria*, 5(2), 1-15.

Montgomery, D.C., Runger, G.C., & Hubele, N.F. (2009). *Engineering statistics*. John Wiley & Sons.

Rotger, B. (1990). *Evaluación Formativa.* Madrid: Editorial Cincel.

Zaragoza, J., Luis Pascual, J.C., & Manrique, J.C. (2009). Experiencias de innovación en docencia universitaria: resultados de la aplicación de Sistemas de evaluación formativa. *Revista de Docencia Universitaria*, 4, 1-33. Available online in: http://cerbero.uab.cat, Moodle Learning Management System Moodle used (last access date: November 2014).

AUTHOR BIOGRAPHY

Carles Sánchez

He has about 5 years of practices teaching experience at graduate level at the engineering school of the UAB. His teachings cover computer science courses like databases and administration of databases on undergraduate level.

Oriol Ramos

He has about 5 years of teaching experience at graduate and postgraduate levels at the engineering school of the UAB. In 2007-2008, he was also a lecturer on the University of Nancy 2. His teachings cover computer science courses like language programming, artificial intelligence and databases, on undergraduate level. At postgraduate level, he has taught on the Master on Computer Vision giving lectures about feature extraction and probabilistic graphical models.

Patricia Márquez

She has about 2 years of teaching experience at undergraduate level at the engineering school of UAB. Her teachings cover science courses like databases and information technology.

Enric Martí

He has more than 25 years experience in teaching Computer Graphics and Databases. He has been coordinator of the Multimedia Technologies Master, and he is the coordinator of the Computer Games Master, both at UAB. He has published more than 25 publications in PBL methodology and in Learning Management Systems (LMS) used in computer science teaching. He has a scientific work published in more than 100 publications, and supervision of PhD theses. His research interests include PBL methodologies, LMS in teaching, computer graphics & interactive systems, augmented reality and serious games.

Jaume Rocarias

He has experience on programming interfaces for undergraduate students focused in personalization of several Moodle environments. He has also given formation courses for PDI staff from several departments.

Debora Gil

She has over ten years of experience at graduate and post-graduate levels. At graduate level, the candidate has taught at the Mathematics (Statistics, Projective Geometry) and Computer Science (Computer Graphics, Programming Languages, Data Structures) Departments of the UAB. Since 2012, she is in charge of the DataBase course within the Bologna grade of Computer Science Engineering. At post-graduate level, she gave a lecture on Optical flow Techniques in the Computer Vision and Artificial Intelligence Master from 2009-2013 and currently cooperates in the definition of the contents of the Graphical Systems and Multimedia course of the Computer Science Master. She has also supervised 12 Master projects (Computer Vision and Artificial Intelligence) and 4 doctoral thesis.