Online Evaluation Methodology of Laboratory Sessions in Computer Science Degrees

Inmaculada Pardines, Marcos Sanchez-Elez, Daniel A. Chaver Martínez, and Jose Ignacio Gómez

Abstract—This paper presents a proposal for assessing the laboratory sessions of a subject of the first year in computer science degrees. This methodology is based on online short-answer exam questions related to the concepts studied in each session. After analyzing the academic results of a large group of students, we may conclude that this way of evaluating knowledge is precise. The obtained grades neither underestimate nor overestimate the student’s work, being similar to the ones achieved in a final exam. Moreover, there is continuous feedback, which allows the teacher to go into detail about those aspects of the subject that students have not understood.

Index Terms—Continuous assessment, online learning environments, teaching/learning strategies, laboratory practical.

I. INTRODUCTION

THE implantation of the new degrees in the European Higher Education Area (EHEA) has brought several changes to the traditional educational model. From the teachers’ point of view, instructors turn into coaches, who help the students follow a suitable learning methodology that allows them to reach a given goal. The students, on the other hand, become the core of the learning process; not only their knowledge but also their skills for achieving it will be evaluated [1]–[4]. Within this context, an efficient ongoing evaluation methodology that assesses the students’ work throughout the academic year (and not only with a final exam) becomes essential. An ongoing evaluation has several advantages, such as forcing students to study and work every day, allowing the teachers to analyze the students’ progress day by day, etc. However, it also generates a significant growth in the teachers’ workload [5], especially in technical degrees, which comprise a high number of lab sessions that must also be evaluated. It therefore becomes essential to develop strategies for rating students completely and objectively; strategies that, at the same time, are simple and automatic enough for keeping the teachers’ workload within reasonable levels.

The aim of this paper is to develop a complete methodology for evaluating the lab sessions of a first-year subject called Introduction to Computer Organization and Design, which belongs to the Computer Science/Engineering degrees taught at University Complutense of Madrid (UCM). We expect such methodology to be reliable, meaning that it can assess students faithfully, something that we will demonstrate by comparing the grades assigned with the ongoing evaluation with those assigned on the final exam; complete, in the sense that it must include the evaluation of all the subject’s contents; and efficient, in that the teachers’ work and the fraction of the lab session assigned to the evaluation are both minimized. Such methodology is based on a short online test at the end of each lab session, that lets both the teacher and the student find out whether the contents of a given lesson have been conveniently comprehended. We will demonstrate through several experiments that our strategy assesses students precisely and completely, while keeping the teachers’ effort at reasonable levels.

Several papers from the literature propose using tests/exams as the basis for an ongoing evaluation, and their results demonstrate that such procedure has a positive effect on the students’ learning process [6]–[12]. Specifically, we have found a bunch of methodologies for evaluating the lab sessions. In [13], the authors propose evaluating students through a project that they must develop, and that is assessed not only by the teacher but also by the rest of the students. Moreover, in [14], a new methodology proposes that students provide teachers with some kind of feedback about the problems and solutions that they found developing the tasks associated with the lab session. The problem with these two proposals is that in both cases the number of students must be low in order not to shoot up the teachers’ workload. Given that in our scenario the amount of students is high, we consider unsuitable employing such methodologies.

In [15] we can find a proposal which is close to the one presented in this paper in several aspects. Like in this case, the practical contents of the subject are divided into several parts or modules. Also, at the end of each module, an exam is carried out to evaluate students and to give them some feedback. A big difference with this proposal is that, in our case, the modules, although related, are not part of a whole project, since that would entail too much complexity, something undesirable for first-year students. Besides, the main goal of our lab sessions is to consolidate the theoretical concepts, which should improve the final results [16].

Finally, we should mention that our methodology makes extensive use of new technologies, for example the Moodle Infrastructure [24], thus significantly simplifying the teachers’ tasks. Several recent studies have proved Information and Communications Technologies (ITCs) to be very useful as a complement to traditional teaching methods [17]–[20].

The paper is structured as follows: the next sections describe how the course is organized and how it is evaluated. The third section briefly describes the methodology applied and explains...
the software interface used from different points of view (students’ and instructors’). A practical case is presented. Finally, the achieved results are analyzed and the main conclusions of this work are presented.

II. Motivation for the Work

The proposed assessment model has been tested in the second semester of the subject Introduction to Computer Organization and Design. This course is taught in the first year of the Software Engineering, Computer Science and Computer Engineering degrees. The subject consists of two semesters clearly distinguished. The first one (6 credits) is focused on computer technology, teaching specification and implementation of digital systems. The second one (6 credits) is an introduction to computer structure, teaching basic concepts of ARM assembler language [21] and the processor and memory system of MIPS computer [22].

This subject is mainly experimental, with sessions distributed throughout the course, trying to consolidate theoretical concepts. Five laboratory sessions of two hours are carried out each semester. In the first one, students have to implement a series of digital circuits. The circuit design has to be developed previously, and in the laboratory session students have to assemble the circuit into the training board and to check its proper functioning. In the second semester, students have to develop a series of codes in ARM assembler language using EmbestIDE environment [23]. They have to program one or two codes before each practical session. Moreover, the codes are checked in the work environment. For this purpose, students have free access to faculty computers or they can install the free version of the tool in their own computers. Through the laboratory session, some changes of these codes are proposed to students, who have to implement them.

Since practical sessions have a significant weight in the final mark of the subject (25% extendable to 40% according to curriculum), they must be completely and objectively graded. Some challenges are presented on this point: Firstly, it is necessary to ensure that the student’s work, both before and during the laboratory session, has been performed individually. Secondly, a wide range of questions covering every aspect relating to the practical lesson has to be formulated in order to obtain an accurate grade of students. However, students have only two hours to be graded, to solve their questions and to develop the proposed changes of the codes. Finally, the high number of students enrolled in the course forces to carry out the laboratory work in different parallel sessions, each one with a different instructor. It is important that instructors do not impose different requirements in each session.

The following methodology is used in both semesters when the proposed method is not applied: at the beginning of the session students are interviewed to check if they have done the design/code individually and if they have understood the associated theoretical concepts. From this brief interview (2 or 3 minutes per student) a first grade is obtained. Then, the student has to implement the circuit in the training board (first semester) or to modify the assembler code (second semester). At the same time any question arising from the implementation is answered. Finally, the teacher checks the proper functioning of circuits/codes and asks some questions to students to complete their grade. Following this methodology when the laboratory is full (20 students) implies that the teacher dedicates over 50% of lab time to assessment.

The proposed methodology tries to make a complete and reliable assessment of practical sessions, releasing teachers of this task (at least during the lab session). This way, instructors can pay special attention to solving questions asked by students and to consolidating theoretical concepts during the practical session. Moreover, it can be expected that the obtained grades of laboratory lessons are a reliable estimation of the final exam assessment since, as we will demonstrate in the Evaluation Section, there is a great similarity between the contents of the final exam and those of the laboratory lessons. Additionally, as it will be demonstrated in the Experimental Results section, the assessment of practical sessions by tests improves slightly the grades obtained in the final exam.

III. Methodology of Lab Assessment

Our proposal is based on the idea of asking students, in each laboratory session, a series of multiple choice questions (or short questions when the multiple choice type is not suitable) related to the practical lesson. The questionnaire is carried out using the educational tool Moodle [24]. Students can use the EmbestIDE environment, where the codes are developed, to answer it. This exam allows us to evaluate the student's handling of the development environment and to check the proper operation of the programmed codes. Moreover, teachers could know if students have acquired the theoretical concepts associated with the practical lesson.

The presented solution organizes laboratory sessions as follows:

1) Propose (7 or 10 days before laboratory session) to carry out an ARM assembly program. Students have to simulate it and check its proper functioning.
2) The teacher proposes students a modification of the original code that they must develop during the lab session.
3) Perform, during the session, a test about the work that has been done.

This methodology presents some advantages:

1) Automatic tests allow to: use self-correction (even in particular questions of short answer), reduce the teacher’s assessment work in the laboratory significantly; give a different, but similar, exam to each student and control the timing of the test.
2) There is also feedback. Teacher scan know easily what concepts have not been understood. Then, they can explain them again during theoretical lessons and even propose additional practical works. On the other hand, students can check if their answers are correct. Furthermore, there is the possibility that they can perform the test again and consolidate the explained concepts. This last feature of the tool is linked to the importance of self-learning methods [25], [26].
3) The methodology is especially suitable for a practical subject:
The test can be done in a reasonable time since it is made up of short questions.

Laboratory assessment is independent of the teachers’ opinion (note that due to the large number of enrolled students in the subject it is necessary to distribute them in different laboratories, each one with a different instructor). Their role will be to help students to understand the concepts associated with each practical lesson and answer their questions.

4) The ongoing assessment improves the student’s knowledge of a subject, as it has been demonstrated in many studies.

IV. GENERATION OF THE LABORATORY ASSESSMENT TEST

Different stages in the process of generating the laboratory assessment tests are described in this section.

A. Question Sets

Each quiz consists of some multiple choice or short answer questions related to the practical work (between 4 and 6 per quiz). Each question is randomly selected by Moodle among a large set (50 or more questions related to the same theoretical concept). This way, there is a low probability that the same test is assigned to several students. Therefore, we assure that the individual work of each student is graded.

Each set of questions is written in a gift format file (one of the importation/exportation data format in Moodle). A line of this file corresponds with a quiz question. The correct answer (or answers) will be after the question text, between curly brackets, preceded by an equal symbol, as it can be seen in Fig. 1.

B. Quiz Generation

Question files have to be uploaded to our online learning environment to generate the quiz. The Moodle tool allows creating quizzes by importing questions from a file. The quiz can be built up from several sets of questions. In this case, different categories (one for each question quiz) must be created. Next, question files have to be imported and linked to the corresponding category.

Using the example of Fig. 2 we describe how to create a quiz with 5 questions. There are 5 categories associated with this quiz, each one with a different number of questions (this number appears in brackets next to the category name).

The following step will be selecting the questions that will define the quiz. Questions will be randomly selected by the Moodle environment. The teacher will choose how many questions of each category will be randomly added to the quiz. He or she will select the category to which each question belongs (see the Category text box in Fig. 2), indicate the number of questions of this category to include in the quiz and click Add (on the bottom right of the figure). Once the total number of questions has been selected, the exam is configured as we can see on the upper left of Fig. 2.

Finally, it will be necessary to edit the quiz to define its visual aspect. We have decided to show one question per page, but students can switch between pages as long as the option Submit all and finish (see Fig. 7) is not chosen. The number of attempts allowed must be also defined; in our tests, we have decided to allow one attempt per question. Finally, as soon as the test is finished, the student will learn his or her final score.

Moreover, we established the start and final time of the quiz, as well as its length according to the time we deem appropriate during the laboratory session. In our experience 30 minutes seems enough to take a full exam, leaving enough time to carry out the modification of the code requested in the laboratory. The editing page to configure these parameters can be seen in Fig. 3.

C. Test Self-Correction

One of the main advantages of performing these tests in Moodle is that the correction work is significantly simplified.

Fig. 1. Example of a question bank (gift format).
This environment has a self-correction system that compares the student’s answer with the acceptable solutions given by the teacher. If they match, the student’s answer is marked as correct and it receives a pre-assigned score; otherwise, it is marked as wrong (in this case, it is possible to give a negative score to the answer). There also exists an option to configure the quiz to show the students their mistakes once the test is closed and provide feedback on every specific question. It may happen that a student provides a right answer whose format does not match the one expected by the teacher. In this case, the self-correction tool would mark this answer as wrong and we would need to solve it through the Item Analysis option in the Results window. In the new window (an example of this option is shown in Fig. 4) all the chosen questions of different quizzes appear along with their expected answer, as provided by the teacher, and a summary of the different answers given by students. For example, in Fig. 4 a question whose answer is “N” is shown; a student has answered “N bit (negative)”, which, being right, has been marked as wrong.

As soon as this problem is identified the teacher has to edit the question to solve it (click icon sited in left column of Fig. 4) and add the new answer as a valid solution. Then, the quiz must be re-graded, so all students that had given this answer will have a new grade. This manually-fixing step would be almost completely removed if Moodle allowed using regular expressions to parse student answers.

Moreover, the Item Analysis window allows the teacher to verify if students have understood the theoretical concepts of the laboratory lesson. If a question has a large number of wrong answers, then the theoretical concepts associated may have not been understood, so the teacher can decide to explain again the corresponding lesson, emphasizing the main objectives of the subject.

Another advantage is that the Moodle environment will grade students automatically after the ending of the quiz if the teacher has assigned a numeric value to each question. Teachers can see the results graphically to know the obtained grades (Fig. 5).

V. PRACTICAL CASE

In this section we describe how our assessment methodology is carried out in a laboratory session. We use the second practice session proposed in the academic year 2011-2012 as an example. With this practice we try to verify:

- Students’ knowledge of the development environment.
- Students’ skill to handle array data structures and their storage in memory.
- Students’ ability to code in ARM assembly language programs specified in high level language.

A practice guide is given to students from 7 to 10 days before the laboratory session. This report reviews the main theoretical concepts of the lab session, specifies the tasks to carry out in the laboratory and explains the operation of the code that students have to develop before the practical session.

The previous work associated with this lab session consists in programming in ARM assembly language the C code
of Fig. 6. This code searches the maximum value among the elements of a positive integer array $A$ of length $N$. Once this value is found it is stored in $\text{max}$ variable. Students are forced to store in memory the $\text{max}$ value each time it changes to generate a large battery of questions regarding memory access. Next, we need to create a large set of questions to evaluate if students have acquired the knowledge and skills required. These are some examples of short questions about the code in Fig. 6:

- Which is the value of $\text{max}$ variable in iteration $i$?
- What value (in hexadecimal notation) is stored at memory address 0x0c00003c after the code is run? (Being 0x0c00003c an address where a variable of the code is stored).
- Which memory address is each element of array $A$ stored at?

In this example, we decided that the quiz was made up of 5 questions, some related to the code programmed at home and some about the work carried out at the laboratory. Each question is associated with a different category. All questions of a category are equivalent in complexity, but each one makes reference to different data. This way, it is guaranteed that all quizzes (there is a test per student) have identical difficulty although questions are chosen randomly.

To increase the number of questions per category and to decrease the possibility that the same question appears several times in different quizzes, we add another strategy. Students are divided randomly and automatically in groups. The same questions are utilized to create the quiz of each group but a different array $A$ is used. In this case, we have to add the answers associated with each array $A$ as correct ones to correct tests automatically. Fig. 7 shows the screen that students can see in the Moodle environment (there are 5 questions, one per page).

VI. RESULTS AND EVALUATION

A. Case Study

We have analyzed the results obtained by 159 students enrolled in the Introduction to Computer Organization and Design course. Students are evaluated through a final exam and also through an assessment of each lab session. To discuss the quality of the proposed methodology, we compare the final exam grades with the lab session ones. The final exam is built by the teachers of the course following a template that has proved its worth over the last years, so this test is the base case for this study. Moreover, in order to enhance the vision we compare the results achieved in the second semester (when the methodology is applied) with the results of the same students in the first semester. Thus, we may evaluate how the methodology affects the same students. However, the first semester is focused on Logic Design while the second is centered on Computer Organization.

Due to the difference between the first and second semesters we also compare, for the second semester, the grades of those students that have followed the methodology (159) with the
results of students that have not followed it (105). These 105 students belong to groups wherein the instructor has followed the same lab session assessment methodology as in the first semester, as was described in Section II.

The previous discussion implies that there are four different grades:
- E1: first-semester grades in the final exam.
- L1: lab-session grades in the first semester (the assessments of this lab session do not follow the methodology presented in this paper).
- E2T: second-semester grades in the final exam for the students that followed the proposed methodology.
- L2T: lab-session grades in the second semester following the assessments proposed in this methodology.
- E2NoT: second-semester grades in the final exam for the students that did not follow the proposed methodology.
- L2NoT: lab-session grades in the second semester (the assessments of this lab session did not follow the methodology presented in this paper).

B. Experimental Results

Table I and Fig. 8 show the statistics of the marks obtained by the students. The reader has to take into account that the highest possible grade for this course is ten points. Particularly, Fig. 8 illustrates through a boxplot the two quartiles above and below the grades’ median.

Table II shows separately the percentage of students that have passed the exam and the lab assessments. Moreover, this table also shows the percentage of students that have obtained remarkable marks (those with seven points or higher).

In order to achieve a more complete conclusion of the experiment we have studied the relationship between the lab-assessment marks and the exam marks for the two groups of students in the second semester (see Figs. 9 and 10). Those figures represent the exam grades (y-axis) versus the lab grades (x-axis) for the two groups of students.
lab-session assessment grades (x-axis). Therefore, with these figures we may quickly visualize the correlation between both grades. Moreover, there are two shaded rectangles that denote the students that have failed the lab assessments and have passed the exam, and vice versa. The lower shaded rectangle encompasses those students who passed the lab assessments and failed the exam. In the case of Fig. 9 (lab assessment without the methodology proposed in this paper) 32.6% of the students passed the lab and failed the exam, whereas in the case of the students that followed the methodology (Fig. 10), this percentage drops to 14.5%. In the same way, the upper shaded rectangle summarizes those students who failed the lab assessments and passed the exam, 11.4% without the methodology and 6.9% with the methodology.

Table III shows the results of the simple linear regression which intends to explain the relationship between the lab assessment and the exam. We have also calculated the Pearson correlation coefficient [27], where 1 is total positive correlation and 0 is no correlation. In addition a t-test [28] is performed, as it can be used to determine if two sets of data (lab-session assessment and exam in this case) are significantly different from each other. A paired samples t-test is applied in the case of the pair exam-lab assessment because they are the same students. An independent samples t-test is performed to compare the results of applying or not the methodology proposed (E2T vs E2NoT).

C. Results Evaluation

One of the first parameters we have studied to check the quality of the methodology is the equivalence between the exam and lab-assessment grades. Table I indicates that the mean (median too) of E2T and L2T, exam and lab respectively, is much more similar than for E1-L1 and E2NoT-L2NoT. We consider the exam as a comprehensive and accurate grade, so that having similar grades demonstrates how effective the lab methodology is. However, one could discuss that a better grade in the lab session than in the exam implies a better use by the students of the lab sessions, but you may also think that the lab session has failed to improve the understanding of the subject.

Moreover, the results showed on Table III confirm that the methodology applied in the lab sessions reaches very close rates to those achieved in the exam (Pearson $= 0.720$). This is not the case for the first semester or for the second semester without the methodology (Pearson 0.392 and 0.300 respectively). Finally, the t-test demonstrates that there are not statistical differences between E2T and L2T (p-value ($T < = t$) higher than 0.05), which means that following the methodology proposed in this paper the instructor could grade the students only through the lab-session assessment. However, there is not statistical relationship between lab assessment and exam for the other cases, the probability is almost zero (p-value ($T < = t$) equals to 1e-31 and 1e-14). This fact should move the instructor to reconsider the methodology to grade the labs.

Those students that passed the lab assessment and failed the exam were over-evaluated (the lower shaded rectangles in Figs. 9 and 10). Maybe these students could have had the feeling that they had a better understanding of the subject than they actually had. On the contrary, those students that have failed the lab assessment and passed the exam were infra-evaluated (the upper shaded rectangles in Figs. 9 and 10). Both situations should happen as rarely as possible, and when they happen, the points in the shaded rectangles should be as close to the center as possible, making the grades from the lab sessions and those from the final exam more coherent.

The results presented not only demonstrate that the lab-assessment grades are more accurate, but with this approach the students are improving the knowledge of the subject. Table II shows that the E2T mean is higher than the E2NoT one, and Table III says that there is a statistical difference among the exam results of these two groups. These two data validate the methodology presented.

VII. Conclusions

The European Higher Education Area has motivated a major shift in the traditional teaching techniques, tipping the balance towards more hands-on based teaching and continuous assessment.

In this work we have introduced a methodology to continuously monitor the progress of first-year students in the Introduction to Computer Organization and Design course. The main goal is to obtain, at the end of the term, a score as objective as it would have been obtained through a final exam. From our results, we may conclude that our methodology measures the knowledge acquired by the students accurately. Moreover, short-answer tests are ideal to avoid subjective teacher interaction and allow evaluating every student individually.

These tests fully covered the topics taught during lectures so the evaluation is complete (lab assignments are designed to complement regular classroom lectures). Last, the teachers’ workload reduces significantly (but for the first year of deploying the methodology) since the test generation and evaluation is fully automatized when using Moodle.

Since students’ final scores for both regular exam and continuous evaluation are very similar, we conclude that we
could safely increase the weight of the continuous evaluation in the global score without over- or infra-scoring students. The methodology largely avoids some undesirable effect of continuous evaluation, when the percentage of students passing continuous evaluation but later failing the exam is too large; it is only around 10% in our case.

Finally we should also stress that students following this methodology obtained better scores than those not using it. We believe that the continuous feedback and the required daily work largely contribute to better consolidating the acquired skills.

ACKNOWLEDGMENTS

The authors would like to thank Guadalupe Miñana, Katzalin Olcoz, Juan Lanchares, Victoria Lopez and María Carmen Molina, for providing data from their assessments.

REFERENCES


Inmaculada Pardines received the Ph.D. degree in physics from the University of Santiago de Compostela, Santiago de Compostela, Spain, in 2007. She is currently an Assistant Professor with the Department of Computer Architecture, Complutense University of Madrid, Madrid, Spain. Her areas of interest are hardware reconfigurable systems. She has authored or co-authored over 20 papers in reviews and conferences. She participates actively in innovative educational projects with the Complutense University of Madrid.

Marcos Sanchez-Elez is currently an Assistant Professor with the Complutense University of Madrid, Madrid, Spain, where he received the Ph.D. degree in 2007. His areas of interest are hardware reconfigurable systems and codesign. He has authored or co-authored over 20 papers in reviews and conferences. He participates actively in innovative educational projects with the Complutense University of Madrid.
Daniel A. Chaver Martínez received the degree in physics from the University of Santiago de Compostela, Santiago de Compostela, Spain, in 1998, and the Electrical Engineering and Ph.D. degrees from the Complutense University of Madrid, Madrid, Spain, in 2000 and 2006, where he is currently an Associate Professor. He has been teaching different subjects related to Computer Science and Electrical Engineering since 2000. His current research interests include architectural techniques for the cache and nonvolatile memories, and OS scheduling for asymmetric multiprocessors.

Jose Ignacio Gómez is currently an Assistant Professor with the Complutense University of Madrid, Madrid, Spain, where he received the Ph.D. degree in 2007. His current research interests include energy-efficient computing in massively parallel architectures with a special focus in the management of the memory system.