Computer Assisted Self-Assessment of Programming Code
A Report on University Students Experience and Opinion

David Insa & Josep Silva
Departamento de Sistemas Informáticos y Computación
Universitat Politècnica de València
E-46022 Valencia, Spain.
{dinsa,jsilva}@dsic.upv.es

Abstract—We have introduced a system called ASys in a university Java programming course. ASys allows students to self-assess their programming assignments, and it also assists teachers when assessing. After the course, we collected the students opinions with a survey. We report on the results obtained and we provide empirical data about the benefits obtained when the assessment made by teachers is assisted by a computer.

I. INTRODUCTION

Assessment is an integral part of instruction, as it determines whether or not the goals of education are being met. When assessment works best, it provides diagnostic feedback both to students and teachers. A special area of interest is the correction of programs written by students because the manual assessment of source code is a time-consuming and error prone task [3], [2], [5], [1]. Even if the assessment criteria are exhaustive, and teachers have test units available to check the source code, finding errors can be very difficult, specially if the project is large, because the teacher has to manually inspect different modules and files. Moreover, if a student submits an exercise whose code does not compile, then no test units can be used to help assessing the code.

There are several other reasons why automatic code assessment is not generalized in university courses. One important reason is that almost all existing systems are based on output comparison with a gold standard. If the output is the expected, the code is correct. Otherwise, it is reported as wrong, even if there is only one typo in the code. Moreover, why it is wrong remains a mystery and quality assessment is not ensured, mainly because it just returns the whole code as correct or the whole code as wrong. No intermediate mark is possible.

In this work we report on the use of a new assessment system called ASys [4] in a real university course. We explain how ASys was introduced in a Java programming course and we describe the experience of both students and teachers.

The rest of the paper is structured as follows: First, in Section II we briefly describe the assessment system ASys. Next, in Section III we explain how ASys was implanted in a real university course. Finally, Section IV concludes.

II. THE ASYS ASSESSMENT SYSTEM

ASys is an assessment system that provides teachers the ability to create exercises that can be self-assessed by students. ASys is also useful for teachers to evaluate students. For instance, given a collection of solutions submitted by students, ASys assists the teacher in the assessment by identifying errors and facilitating their automatic correction and marking. Specific information about ASys can be found here:

http://users.dsic.upv.es/~jsilva/ASys/

ASys is particularly interesting because it goes beyond output comparison, and it can assess unrestrained code. In particular, it can automatically assess the code not only from the final output, but also checking whether the own source code fulfils any properties desired by the teacher (e.g., definition of a particular class hierarchy, implementation of interfaces, existence of a particular field, etc.), thus allowing for marking the code even if it is only partially correct. Whenever human intervention is needed, the system prompts the teacher with the available information showing the problem found, and both the student’s code and the solution.

Example 2.1: In order to solve a Java programming exercise, a student must declare a class Cube that extends class Figure. The student failed to fulfil this requirement. Fortunately, ASys can automatically identify this problem. If a teacher is assessing the exercise, then ASys is able to automatically correct the problem, recompile, test the solution and assign the appropriate mark. If a student is self-assessing the exercise, then ASys just provides a tip, so that the student can solve the problem on his own.

Figure 1 shows a screenshot of ASys where the problem is specified at the top, the student’s wrong code is shown inside the left panel, and the correct code is shown inside the right panel. At the bottom, it shows what is the value assigned in the assessment template to this error, and it asks the teacher what mark should be finally assigned.

Together with the corresponding mark, the teacher can also include a comment associated to each error. This is very useful for the student, who will receive a report with his original code and the corrections made by the teacher (including the
comment associated to each error). These comments are also useful for teachers, because they can share the comments and marks associated to the same error made by different students. In particular, ASys maintains a database of triples (error–comment–mark). When an error is found, and it is already stored in the database, ASys suggests the associated mark and comment to the teacher. In this way, the same criteria is used for different students, and also in different groups (by different teachers).

III. THE EXPERIENCE: ASYS IN ACTION

We used ASys in a real university programming course with a double aim:

1) Assisting students, who used the tool to self-assess exercises prior to the exam, and
2) assisting teachers (different from us), who used the tool to automatically assess the final exam.

5 teachers and 535 students participated in this experience. A. ASys used by students

Prior to the course, we created six self-assessing exercises with ASys. Three of them were exams of the previous year. Along the course, the students had available the student version of ASys together with the six exercises, and they were free to use ASys to prepare the final exam. 82% of the students used ASys.

What ASys provided to the students is the possibility to detect any compilation error and any evaluable property that was not satisfied by their code. Moreover, together with each error alert, ASys provided tips to solve it. In this way, when ASys reported their code as valid, the students were sure that their code was functionally equivalent to the teachers solution with respect to the exercise requirements. This is the most valuable contribution of ASys, because previous years the students also had exercises available but, without ASys, after solving them, they could not be sure that their solution was correct.

After the course, we conducted a survey to precisely know the students opinion about the different aspects of ASys.
Figure 2 summarizes the main questions of the survey.

From this information we can conclude that students found ASys a very useful resource. It is interesting to notice that ASys solved many of the problems of the students and, thus, they asked less questions to the teacher, who could concentrate on the really important problems. It is also remarkable the fact that ASys by itself is not enough to pass the subject according to the students perception.

B. ASys used by teachers

ASys allows to automatically mark exams if we base the marking criteria exclusively on testing. In these subjects, this was not the desired option, because many functional and non-functional (such as maintainability of the code) requirements were taken into account when marking. Hence, the exams were assessed using the semi-automatic assessing mode of ASys. In this mode, ASys identifies errors, but it is the teacher who decides the mark. In total, 6 exams were performed and assessed with ASys along the course.

To study the impact of ASys on the assessment, all teachers assessed the exams twice. First, they assessed the exams manually (as usual), so that we could know the final marks without the use of ASys. Then, they assessed the exams again with ASys. The first important conclusion of this experience is that ASys detected many errors that were undetected in the manual assessment. The average error made by teachers when assessing was +0.15 points out of 10 (+2.61%). Assessing errors in the manual assessment were 75% of the times positive (they benefitted the student) and 25% of the times negative (they harmed the student). In many exams there were various errors, some of them positive and others negative that compensated theirselves. The average cumulated error per exam in the manual assessment was $|0.49|$ points out of 10 (4.9%).

To complement the empirical study, we also conducted several interviews with the teachers. They found the tool useful for three reasons: First, because it saves a lot of time in the assessment. Second, because it increases the autonomy of the students, and thus they received many less questions about how to solve the exercises. And third, because it increases the assessment quality, not only because it is exhaustive, but also because it is completely fair and equally applied to all students.

IV. Conclusion

This paper reports about our experience using the automatic assessment system ASys in a Java programming course. The results obtained and the information collected shows a clear benefit of using this kind of tools. In particular, after the course, more than 90% of the students found ASys to be a useful resource. In fact, most of them formally asked for ASys in the following programming course focussed on Haskell.

From the teacher’s perspective, ASys is a rigorous method that ensures quality in the assessment. It is exhaustive, fair and quick. For this reason, the teachers that participated in the study have officially integrated ASys in the subject for the following courses.

ACKNOWLEDGMENT

This work has been partially supported by the EU (FEDER) and the Spanish Ministerio de Economía y Competitividad under grant TIN2013-44742-C4-1-R, by the Generalitat Valenciana under grant PROMETEO-II/2015/013 (SmartLogic), by the UPV’s Instituto de Ciencias de la Educación (ICE) under grant PIME 2015-B16, and by the UPV’s Escuela Técnica Superior de Ingeniería Informática.

REFERENCES