Self-Assessment Web Tool for Java Programming

Bruno Baruque and Álvaro Herrero

Abstract Self-assessment capabilities, that enables the student to have an insight on his own learning process, are a very desirable skill on any higher education student. It is even more important for transnational students as it allows them to successfully adapt to their new international learning environment. Present work proposes an online tool to help computer science students to develop their self-assessment skills while learning Java programming. It consists on a plug-in for the widely popular Moodle learning management system to work with source code for the Java programming language. The developed plug-in lets students to upload source files, analyzes the code and presents a report to the student using industry standard tools. The report includes both errors, points of improvement and a general comparison with the rest of his classmates, from the software quality standpoint. This way, the student is provided with a framework against to which compare the correctness of the solution he/she has programmed before it is delivered to the teacher for evaluation. The teacher is able to access all this information too, facilitating an insight of how the class, as well as individual students, is progressing. It is expected that the shortening on the classical loop of the student problem solution and teacher feedback will enhance the student self-awareness and will improve his overall performance in programming courses.

1 Introduction

Formative assessment can be defined as employing appropriate activities to provide feedback to enhance student motivation and achievement during instruction—as students learn. Providing helpful information while learning occurs contrasts with providing feedback solely after instruction (examination).

B. Baruque (✉) · Á. Herrero
Civil Engineering Department, University of Burgos, Burgos, Spain
e-mail: bbaruque@ubu.es

Á. Herrero
e-mail: ahcosio@ubu.es

Á. Herrero et al. (eds.), *International Joint Conference*, Advances in Intelligent Systems and Computing 369, DOI 10.1007/978-3-319-19713-5_51

According to [6]

> "Self-assessment is more accurately defined as a process by which students (1) monitor and evaluate the quality of their thinking and behavior when learning and (2) identify strategies that improve their understanding and skills. That is, self-assessment occurs when students judge their own work to improve performance as they identify discrepancies between current and desired performance."

This ability of being capable to understand how learning is taking place and what can oneself do to progress in this task is very valuable for everyone, but specially for people on their formative years. This can be seen as a crucial ability for students of higher education and more precisely, those pursuing STEM-related (science, technology, engineering and mathematics) degrees [3]. This kind of studies are characterized for being rich in complex problem-solving tasks involving also abstract concepts and multiple ways of tackling the same problem. This has been recognized by the European Higher Education Area (EHEA), which stresses the need for students to develop key cross-curricular skills such as "problem solving", "information management" or "critical thinking" [4].

This is especially true in computer science studies, where students are taught how to build automated solutions to real-life problems or to manage already existing complex systems to perform these tasks on a daily basis. This area of knowledge is particularly prone to this kind of self-assessment activities, since usually students can test whether the solution they have designed works or not by running a test on a regular computer system. Present work focuses on computer programming courses, but the underlying principles could be extended to other IT disciplines.

Usually, programming practical lessons follow a similar scheme: the teacher proposes a problem which will be solved by writing code in a certain programming language. A very well defined set of requisites is delivered to students for their solutions to comply with. Students then design an algorithm needed for the solution and implement it. After that, the teacher collects all the proposed solutions and checks them (according to the requisites). Finally, feedback is given to the students (probably along with a numerical mark) by the teacher.

Since students in this area—and increasingly in many others nowadays—are used to have access to interactive ways of learning, it seems quite natural to have a system to guide them when completing the proposed tasks, rather than waiting for the teacher to either give direct feedback during class sessions or ultimately mark their exercises to assess the quality of their work.

To overcome this limitation, the aim of the proposed system is to automate the feedback loop for the students and therefore reducing the time frame when they receive assessment on the quality of their work. This would make it more flexible, as students do not need to be in the same classroom as the teacher. It will also add a way to compare their work with that of their classmates in order to provide a context to reinforce the self-awareness of the students [8, 10]. The main characteristics of the proposed web tool are described in the following sections.

2 Related Work

There are many other efforts being developed and tested in order to help university students master complex concepts through the help of technology [10]. This trend has been even emphasized in later years, with the wake of MOOCs (Massive Open Online Courses) systems, that provide access to courses of different disciplines to hundreds or even thousands of students at the same time.

Some of those systems are used for disciplines such as Physics [11], Chemistry [2] or Mathematics [3], among others. There are other ones that are aimed at lowering the input barrier for beginner students to programming courses [12]. Some efforts even go one step further, creating adaptive paths of exercises for students according to their progress on previous tasks in the platform [5].

In present case, the proposed tool has some specific characteristics that make it different from the others. First of all, the tool is aimed at students who have mid-level expertise in programing. That means that, probably, they have already mastered the basics of the topic and are learning more advanced concepts. Therefore, the output of the tool is based on industrial standards, rather than simplifying the information for everyone to understand. That way, students start to relate with the outputs offered from professional tools, while they are shielded to a certain degree from all its complexity. Additionally, it is specifically programmed to enable the tasks to be completed by a group of students, automatically sharing the results of a test with all the group members. This is meant to encourage collaborative work among students. Another advantage of the proposed tool is that it enables students to compare their performance with the rest of the course (in an aggregated anonymized way). Again, this feature presumes some maturity from students in order to interpret the obtained results. This intends to reinforce the aspect of self-assessment, indicating if the student should make an effort to reach the same progress state where the rest of students are or not.

3 Proposed Automated Self-Assessment Web Plug-In

3.1 Objectives

The main objectives of the tool designed for each student to perform self-assessment are:

From the teacher point of view: The tool must be able to detect and highlight errors or malfunctions in the code written by the student. It must be flexible enough to let the teacher decide which are the errors that he/she wants to highlight for the students, including the feedback given to the students. It would be also interesting to spot other errors in an automated way, to help the teacher with the task.

The tool should be able to present the teacher different levels of information detail about the assignments. The tool should present the teacher with an overview of the

classroom progress per assignments. It should be able to objectively measure this according to standard measures of software quality. It should also be able to disaggregate this information for different assignments and individual students (or group of students).

From the student point of view: The information presented to the student should be clear and informative. The process of self-assessment should be as straightforward as possible for the student. The student should see the application as a helpful tool for his learning and not the other way round.

The students should see the tool as a help and guide for his own self-assessment, but let him complete the progress by himself. It is important for the student to internalize the process of reflecting about his work, finding errors or weak spots in his reasoning... as a summary, to let him engage in the problem-solving dynamic. The tool should not appear to him as a trial-and-error way of completing the assignments.

The tool will also let the students (or group of students) to compare his progress to the rest of his classmates.

3.2 Tool Description

The tool is implemented as an extension of the well-known Moodle Learning Management System [7]. This has mainly two advantages. Firstly, as it is one of the most widely spread LMS, both teacher and students are probably familiar with the basic workflow of the tool. Second, the user authentication system, assignment setting and many other aspects of LMS are already implemented, letting us focus on the specific task of programming self-assessment.

The software is then programmed using standard languages and tools for web programming (PHP and MySQL mainly). It is divided into several modules, which relate to different aspects of the teaching-learning management. In present work, authors have extended the module of "Tasks" (specifically the "assignment-upload"), which is the part that implements the workflow for delivering assignments to be graded by the teacher.

What the extension does is altering this basic structure by performing some automated actions when the student uploads the source code for the assignment, providing both the student and the teacher with complete information about the quality of the proposed solutions. Up to now, the extension is devised for the assessment of programming assignments implemented in the Java programming language. In order to do so, it relies on the results obtained from two of the most used libraries for testing the correctness and quality of source code: jUnit [13] and PMD [9]. The first is used to check the functionality of the program proposed by the student (unitary tests), while the second one is used to analyze problems or suggest best practices on the codification of the program (static code tests). The use of JUnit implies that it is the teacher the one responsible for defining the exact behavior that is required on the students program and the feedback that will be given to them when the behavior presented does not match the requested specifications. Students will see the task as

any other upload assignment task in the course. The difference is that, as soon as they upload their answer (implemented as Java source code); they will be provided with an exhaustive feedback automatically obtained from the teacher's jUnit tests and PMD analysis results. This process is limited to a certain number of tries (set by the teacher), in order to prevent students to engage in a trial-and-error process; rather than a reasoned solution to the weakest aspects of the assignment.

As depicted in Fig. 1, the typical workflow for using the proposed tool would be the following:

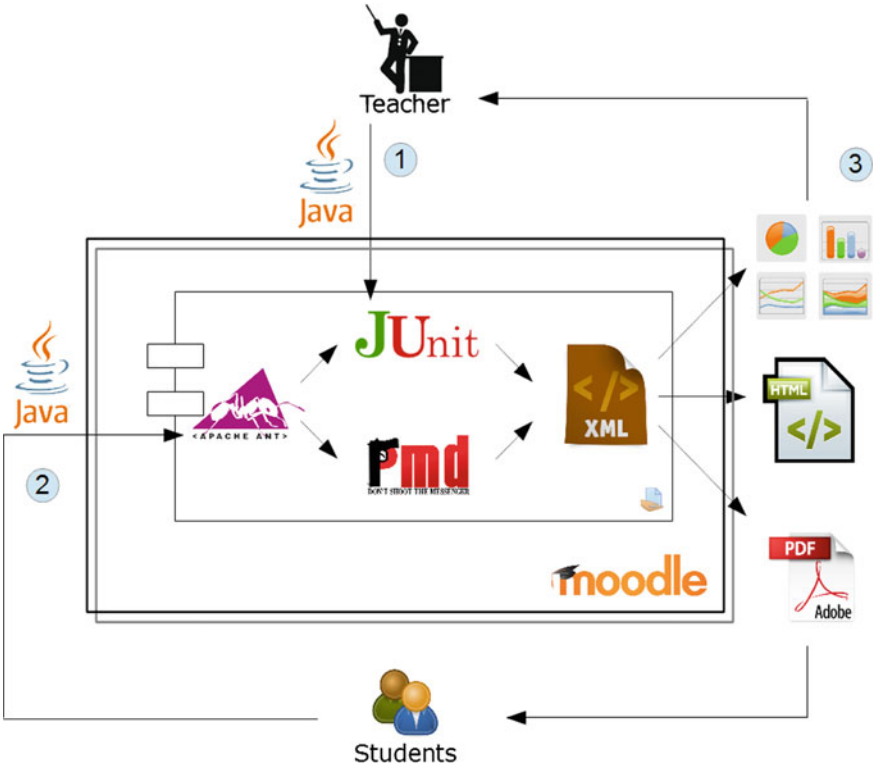

Fig. 1 User (teacher/students) interaction with the proposed system

1. The teacher designs a new programming assignment and prepares the jUnit tests that will be used to check its correctness along with its description; in the way Test Driven Development (TDD) [1] is performed in the industry.
2. The teacher creates a new programming assignment task in the Moodle system, and sets the task as usually. For this task, a suite of jUnit tests is also required. Additionally, the maximum number of tries for each student to use the auto-correct tool can be set. In Fig. 1, this is labeled as "1".

3. The students works on the assignment and finally get the requested program. They are advised to do their own tests while working on the assignment. When they think the program is complete enough an want to check it, they can upload a copy into the correspondent task space (as a zip file). In Fig. 1, this is labeled as "2".
4. The upload is analyzed in the server by using the ANT [14] library to automatically perform the tasks of compiling, running tests, saving the results, calculating statistics and formatting all the information to be presented to the user (teacher and student). If the uploaded files complies with some rules and can be compiled correctly, a summary of the errors and/or advise to improve their program are shown to the users. The amount of details for this feedback is set by the teacher when creating the task. In Fig. 1, this is labeled as "3".

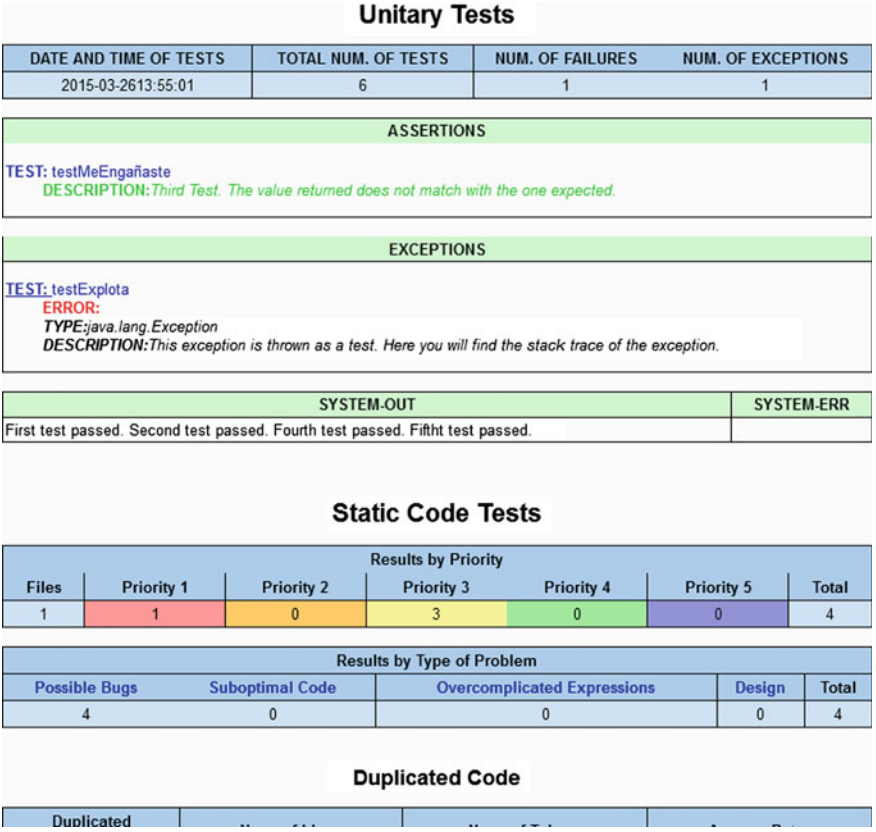

Fig. 2 Overview of the results obtained from the unitary and static code tests that are displayed to students. The user can select any group of errors for further inspection. Automatic feedback for each individual error, included by the teacher, is also shown (on demand) to the student

5. Students can visualize both an overview of results or each one of them individually, for further details. A high-level comparison with the rest of the classmates is also offered to students, comparing the errors of their assignment with an average of the class. They can choose to read this on the Moodle web application or download a report in PDF format for later analysis (Figs. 2 and 3).

While students are uploading their answers, the teacher can access at any time to a summary or snapshot of how the whole class is working on the assignment, presented as an aggregation of the individual errors from each one of the students. For a deeper knowledge of students performance, the teacher is able to select a given student and analyze in detail the source code uploaded and the exact errors produced. To do this, access to both the student uploaded file (with the source code) and the same report the student has received as feedback (on web and PDF form) is given

Fig. 3 Overview of the summary of errors of the whole class for a given task, when accessed by the teacher. Details about how many errors each group of the class has is also included in the data shown. The teacher can also obtain details about each individual error in the delivery of each student/group

to the teacher. This is a great insight into the progress of the class and can be used to prepare the next class sessions to better help their students by focusing on those issues where students are having problems.

4 Planned Experiences and Expected Results

The presented software is a completely functional tool, that has passed all software testing needed in this type of software. For the first field testing to be performed, the tool will be used to assist students with the assignments of programming classes on a second-year subject from the bachelor studies in the Computer Science Degree at the University of Burgos. This subject is one of the most difficult ones for students, since it requires them to master complex concepts and students are still not proficient enough with their programming skills (although they have already completed two basic programming subjects). It is expected that the tool will help them complete their assignments by informing about their errors before they deliver the final code for marking, so they can reflect on why their program is failing and how to correct these errors. This should improve over the usual dynamic in which the students receive the feedback on their work at the same time of their marks and have already moved on to complete the next task, so very little reflection is done over the flaws and errors of previously completed tasks.

The on-line approach of this tool makes it especially suitable to be used by the students from the on-line Computer Science bachelor degree already being implemented at the University of Burgos. Since the interaction of the group of students with the teacher becomes more reduced in this type of studies, the advantage of having a mean to automatically assess about the correctness of the tasks in a more flexible way than e-mail or forums should be quite appreciated.

Finally, the proposed web application is quite convenient for international/transnational students. Moodle is a tool that has been translated into many different languages. Following the same approach, the developed auto-assessment tool is also prepared to be easily translated into any language (already including English and Spanish). This would be very easily integrated in courses that require student to move between different countries, as it can be used to help them follow courses taught in a different location with less effort. If a student finds it difficult to follow the teacher of understand some of the explanations because of the language, it will be much easier for him/her to work with the developed tool.

This can help also to incoming international students. For future incoming students is easy to provide them with a sample of the work local students are required to complete at a certain moment (with the exact same exercises and feedback). For current students, having the choice to receive feedback using their own language with very low effort (or practice a different one) could also be considered an interesting advantage.

5 Future Work

The system is regarded as a very interesting tool for supporting any programming assignment to university students. This means that it may not be the best way of introducing programming to inexperienced students, since it assumes students are familiar with reading requisites and complying with strict programming rules; but it could fit very well with undergraduate students from their second course on to the final courses.

As this was the initial setup for the software, it shapes a bit the main improvements and future work to be done in the system. Up to this moment, the system is not prepared to cope with problems mainly raised with students in their very first programming tasks: infinite loops, compiling problems, very sub-efficient solutions (time and resource wise); that could affect the performance of the system and even blocking access to other users.

Also, as versions of the Moodle system and the other tools (jUnit, PMD, ANT, etc.) are released regularly, the software should be updated, taking into account the requisites that newer versions of Moodle imposes to its modules. At the same time, the latest Moodle versions give the solution for this same problem: Moodle currently complies with the Learning Tools Interoperability (LTI) v.1.1 standard. This means that it includes several modules that enable Moodle to communicate with any other web tool that implements this specification. This way, both pieces of software can evolve at their own pace, maintaining their functionality intact. One of the clearest ways of improving the proposed auto-assessment system would be to implement the system in its own, dedicated server.

Finally, the ultimate goal of the system is not only to help students with the assessment of their work in a rather static way, but also to analyze their answers to different proposed task and guide them into new problems that could help them master concepts that are still missing from their learning, creating a personalized pathway through their programming courses, maximizing their engagement and learning outcomes.

References

1. Beck, K.: Test driven development: by example. Addison-Wesley Longman, London (2002)
2. Dalgarno, B., Bishop, A.G., Adlong Jr, W., Bedgood, D.R.: Effectiveness of a virtual laboratory as a preparatory resource for distance education chemistry students. Comput. Educ. **53**, 853–865 (2009)
3. Hotard, D.J.: The effects of self-assessment on student learning of mathematics. Ph.D. thesis, Louisiana State University (2010)
4. Keraudren, P.: New skills and jobs in europe: pathways towards full employment. Technical report, European Comission—Directorate-General for Research and Innovation (2012)
5. Marcus, N., Ben-Naim, D., Bain, M.: Instructional support for teachers and guided feedback for students in an adaptive elearning environment. In: Eighth International Conference on Information Technology: New Generations (ITNG), pp. 626–631 (2011)

6. McMillan, J.H., Hearn, J.: Student self-assessment: the key to stronger student motivation and higher achievement. Educ. Horiz. **87**(1), 40–49 (2008)
7. Moodle Pty Ltd: Moodle web site. https://moodle.org/. Accessed 15 Jan 2015
8. Panader, E., Alonso-Tapia, J.: Self-assessment: theoretical and practical connotations. when it happens, how is it acquired and what to do to develop it in our students. Electr. J. Res. Educ. Psychol. **11**(2)(30), 551–576 (2013), http://dx.doi.org/10.14204/ejrep.30.12200
9. Pelisse, R., Dangel, A.: PMD web site, http://pmd.sourceforge.net/. Accessed 15 Jan 2015
10. Pirker, J., Riffnaller-Schiefer, M., Gütl, C.: Motivational active learning: Engaging university students in computer science education. In: ITiCSE '14, Proceedings of the 2014 Conference on Innovation and Technology in Computer Science Education. pp. 297–302. ACM, New York (2014), http://doi.acm.org/10.1145/2591708.2591750
11. Prusty, G.B., Russell, C., Ford, R., Ben-Naim, D., Ho, S., Vrcelj, Z., Marcus, N., McCarthy, T., Goldfinch, T., Ojeda, R., Gardner, A., Molyneaux, T., Hadgraft, R.: Using intelligent tutoring systems in mechanics courses, a community of practice approach. In: The 22nd Annual Australasian Association of Engineering Education (2011)
12. Ramos, J., Trenas, M.A., Gutiérrez, E., Romero, S.: E-assessment of matlab assignments in moodle: application to an introductory programming course for engineers. Comput. Appl. Eng. Educ. **21**(4), 728–736 (2013), http://dx.doi.org/10.1002/cae.20520
13. Saff, D., Cooney, K., Birkner, S., Philipp, M.: jUnit web site. http://junit.org/. Accessed 15 Jan 2015
14. The apache software foundation: the apache ANT project. http://ant.apache.org/. Accessed 15 Jan 2015