Investigation of Continuous Assessment of Correctness in Introductory Programming

Deller James Ferreira[1*], Hebert Coelho da Silva[1], Tatiane F. N. Melo[2] and Ana Paula Ambrósio[1]

[1]Instituto de Informática, Universidade Federal Goiás, Câmpus Samambaia, Goiânia, Goiás, Brazil // [2] Instituto de Matemática e Estatística, Universidade Federal Goiás, Câmpus Samambaia, Goiânia, Goiás, Brazil // deller@inf.ufg.br // hebert@inf.ufg.br // tmelo@mat.ufg.br // apaula@inf.ufg.br

[*]Corresponding author

(Submitted March 9, 2016; Revised August 4, 2016; Accepted October 25, 2016)

ABSTRACT

Teachers usually expect that any form of a continuous assessment (CA) should contribute significantly to the student learning process in introductory programming courses. To foster teachers to go beyond the current practices of a CA applied when to programming, from 2011 to 2014, we investigated the use of the Boca Online Contest Administrator (BOCA) system, an online judge used in programming marathons, and the CA of program correctness in the cohorts of an introductory programming course. Empirical results show that there is no significant difference between student's performance when comparing the cohorts that used a CA and did not use a CA, and when comparing the cohorts that used and did not used the BOCA system but used a CA. An in-depth analysis revealed the potential and limitations of the use of a CA and the BOCA system, unveiling the need for the adoption of assessment practices and environments that build cumulative knowledge through multiple means of assessment, allowing profitable interactions among students, and between students and their teacher regarding the students' solutions.

Keywords

BOCA system, Continuous assessment, Introductory programming

Introduction

Learning how to program is notoriously complex and difficult. Computer educators share the assumption that their students find programming difficult to learn (Jenkins, 2002; Bornat, 2011; Dijkstra, 1982). In addition, programming demands several skills that are intertwined, and the novice programmer needs to deal simultaneously with multiple processes.

Solutions to computer programming problems should be modular, efficient, clear, validated, and accurate. In addition to being a creative activity, students have to learn the syntax and semantics of programming language elements, and how to combine them into meaningful programs (Muller, Haberman & Averbuck, 2004; Robins, Rountree & Rountree, 2003; Utting et al., 2013). To succeed, students must develop the ability to deal with computational problems regarding the many creative and cognitive aspects of programming.

In addition to difficulties in programming, student performance is frequently below the teacher's expectations (McCraken et al., 2001). For example, Jerinić et al. (2014) assessed the grades of students taking introductory programming at different institutions and in various countries, and concluded that the students in introductory programming courses do not know how to program at the expected skill level.

Many researchers have considered different perspectives to the teaching and learning of programming (Koulouri, Lauria & Macredie, 2014; Soloway & Ehrlich, 1984; Sphorer & Soloway, 1986; Tang, 2009; Denadhi, 2009). However, there is still no revolutionary pedagogy to teaching programming, and no consensus on what is the best way to learn it. Meanwhile, most teachers still apply a traditional teaching method for introductory programming courses that consists of lectures and programming exercises.

There is empirical evidence that programming exercises can have an even more important role than simply applying the theory taught during lectures. Exercises can be seen as teaching artifacts that complement lectures by teaching the same content but in an exploratory manner.

In an attempt to make the learning of programming more effective, feedback should be provided to the students during their programming exercises because they learn more effectively when they know what is expected from them. Students look forward to feedback that enables them to improve as learners. During programming activities, feedback is often given by means of assessment tasks (McCraken et al., 2001; Earl & Uscher, 2012).

An assessment is any act of interpreting or acting upon information regarding a student's performance, which is collected by a variety of means or practices (Nicol & Macfarlane-Dick, 2006; Messick, 1989). An assessment can significantly influence the effectiveness of student learning (Hattie & Timperley, 2007; Vihavainen, Airaksinen & Watson, 2014). Educators should consider applying assessments as an integral and important component of the teaching and learning process.

For the purpose of boosting student learning, a CA is seen as an ongoing process arising out of the interactions between teaching and learning. An assessment involves both the teacher's and the student's use of information, and provides evidence of student performance. The main aspect of the assessment process is to enhance an evaluation by means of a CA that can imply the student's level of progress (Fisher & Frey, 2007). A CA can help a student become more aware of any gaps that exist in their learning process, and can motivate them to narrow these gaps before taking an exam.

Feedback can offer students an experiential base for reflection. For the purposes of this work, we consider reflection as a mental process that incorporates critical thought regarding an experience. A student's ability to reflect on the materials they have produced, in order to form reasoned judgments, is central to a deeper level of learning. An assessment should provide insight to the students to enhance their learning processes by allowing them to identify their strengths and weaknesses. By means of such reflection, the students can practice and demonstrate self-knowledge to create a new and different course of action.

However, assessing the students and providing appropriate feedback to them in terms of their programming are not simple tasks. Some educators examine and grade their students' assignments manually, whereas others prefer automatic tools to ease the efforts required for the assessment. Automatic programming assessment systems have recently become a significant method in assisting educators to automatically assess and grade their students' programming exercises. Among these tools is the Boca Online Contest Administrator (BOCA) system (de Campos & Ferreira, 2004).

Many researchers have highlighted the evidence indicating that an automatic programming assessment can provide data to increase student performance. These data are directly related to the criteria chosen to assess a computer program. Given a particular problem to be solved by the students, there are a number of criteria that can be used to assess the computer program used to solve the problem.

An automatic programming assessment system can assess whether a program is correct. In addition to correctness, some assessment tools are used to analyze the program efficiency, coding style, and the existence of inline documentation. However, there are other important criteria used to judge the level of programming. Because programming is creative and cognitively complex, and because there are constraints imposed by the programming language used, it is important for the student to receive data regarding the act of programming, and not only data judging the program itself. The use of automatic tools focuses mainly on the correctness of the solutions and neglects other important properties such as the design and clarity of the program.

For example, the students need to know whether the chosen data structure is appropriate, whether the chosen control flow is the best, how to structure the code used to solve a problem into smaller units (which can potentially be re-used), and what algorithm or programming patterns are going to be applied, combined, and adapted. For both the students and teachers, the variety of different correct solutions can be a challenge.

In addition, an international study on introductory computer science courses conducted by Carter et al. (2010) showed that most teachers assess the programming exercises submitted by their students merely for their correctness, either manually or using an automatic assessment tool.

Thus, considering that certain works highlight the need to embrace criteria other than the correctness during a CA of the programming level, and being aware that providing only feedback regarding the correctness of a program is a current practice adopted by teachers, further research that aims to provide a deeper understanding about what an assessment of correctness is, and what this type of feedback can obtain from the students regarding their achievements, is essential. Capturing and highlighting the educational benefits and limitations of an assessment of the correctness in programming are required to better enlighten the teachers.

In this work, we investigate the continuous assessment of correctness in introductory programming. The results indicate that, despite the correctness of their code being an essential aspect of a programmer's work, an assessment of the correctness provides important but insufficient data for effective feedback in promoting student reflections and in helping teachers develop facilitation strategies.

Related works

Teaching can be considered a closed-loop feedback process, where the teachers require an achievement measure of their students' learning to be aware of the weaknesses and strengths of their students, and to overcome such weaknesses (Ashour et al., 2014). Clearly, an assessment applied during the learning process is valuable, and any evaluative works conducted prior to the teaching are deemed worthwhile.

According to Serra-Toro, Traver, and Amengual (2014) despite continuous assessments being commonly used in higher education, some of their limitations are often ignored. Continuous assessments can influence the ways in which students approach their studies, and therefore contribute indirectly, but effectively, to the quality of their learning, but only when carefully designed.

Nevertheless, the relationship between assessment practices and the overall quality of teaching and learning is still undermined. There is often a lack of assessment requirements and assessment criteria assuring the effectiveness (Brown & Hattie, 2012).

According to Hattie and Timperley (2007), the type of feedback can cause a greater or lesser degree of effectiveness. Feedback in programming can even fail, as reported through previous research (Soloway & Ehrlich, 1984; Scott et al., 2015). Scott et al. (2015) observed that continuous feedback is not always efficacious, depending on certain factors such as the amount, nature, and timing of the feedback.

The continuous characteristic of the assessment does not guarantee student progress. Successful continuous assessments must display evidence regarding the level of learning at a sufficiently early time period, in such a way that the teaching and learning can be modified, causing greater progress toward the intended goals and targets. Progress involves being able to perform faster, more accurately, and more easily; knowing more; and achieving a deeper programming capability (Scriven, 1967).

An adequate measurement of a student's learning performance should provide opportunities for improvement, along with guidance on the redesign of the course objectives. Assessments applied to improve the degree of learning must be used to diagnose an individual's learning difficulties and suggest remediation procedures.

Ihantola, Karavirta, and Seppälä (2011) made a systematic literature review regarding systems for an automatic assessment of programming assignments. They indicated that an automatic assessment has certain limitations. One of the limitations of current automatic assessment tools, such as the BOCA system, is the limited quality of the automatic feedback generated.

The use of automatic testing allows the students to be accurate and obtain the maximum score, but does not guarantee that several of the program requirements are checked. Automatic tools cannot examine whether a variable name is meaningful, if the program is well designed, or even if the program uses a specific data structure. Automatic tools focus mainly on checking the solutions, and neglect other important aspects. In addition, the final product (program) tends to be emphasized over the process (programming).

A CA, whether automatic or not, can contribute to the student learning process only if the data used for the feedback, or used to generate the feedback, come from a set of relevant assessment criteria, including not only the criteria for a program assessment, but also the criteria for judging the act of programming itself.

Implementing a CA properly requires the teachers to reconsider their role, their students' roles, and their interactions with their students. Interactive techniques for a description and identification of the causes of programming errors need to be applied more frequently. Teachers need to watch the behavior of their students while programming, and listen closely to their conversations. At times, they may need to ask questions during conversations to clarify details regarding what the students are doing and what they are discovering, but otherwise, they should not interfere with the students. Teachers need to ask purposeful questions that enable the students to reflect upon, clarify, and explain their thinking and actions. In addition, it is profitable to immerse the students in collaborative learning, where student dialogue is valued.

An investigation into the application of a CA of correctness, either using an automatic programming assessment system or not, is a relevant contribution. To delimit the influence of providing feedback regarding the correctness of a program for the learning of novice students, revealing the potentialities and limitations of the program is a necessary issue to motivate teachers into incorporating interactive assessment procedures and feedback strategies to assure the quality of learning how to program.

Research questions

To reveal the limitations and potentialities of a CA of correctness in introductory programming, we conducted a comparative analysis by considering the cohorts when the BOCA automatic programming assessment system and a manual CA of the program correctness were and were not applied, with the objective of enlightening and encouraging teachers to go beyond adopting such instructional practices. The research questions applied are as follows:
RQ1. Do students in cohorts, where the teacher applies continuous assessment and the programs are checked for correctness, attain better educational achievements?

RQ2. Do students in cohorts, where the teacher applies the use of BOCA for program assessment, reach better educational achievements?

Method

A comparative study was conducted with the goal to investigate the use of the BOCA system and a CA of correctness in an introductory programming course. To answer the research questions, an empirical study was conducted using statistical techniques to analyze the results obtained from an assessment of the students' performance.

The BOCA system is an automatic program correction system commonly used in Brazil for checking programs using predefined inputs and outputs. The BOCA system receives the students' submissions and uses the given inputs to verify whether a program's output is identical to the expected output for the given input. The BOCA system serves as a repository of solutions submitted by the students, where feedback regarding the correctness is provided.

Quantitative measures for analyzing data have been addressed, and empirical methods have been employed. An empirical method reveals observational data, making it possible to make inferences considering the effects of the use of the BOCA system and the application of a CA, and to conduct a comparative analysis between cohorts in which the BOCA system and a CA were and were not applied.

The statistical techniques used to compare the different educational situations relevant to the above research questions are the Quantile-Quantile (QQPlot) graph and the Mann-Whitney test. The empirical data measurements used were the final exam grades and ongoing assessment testing grades.

QQPlot generates a graph used to compare the characteristics of two populations, and allows a very intuitive visualization to check whether there are differences between the grades of the cohorts. If there are no significant differences between the grades, the dots are close to the diagonal line, inclined at 45º, passing through the origin. The Shapiro-Wilk test showed the non-normality of our data at the 5% level, which lead us to opt for the Mann-Whitney (*U)* test. The Mann-Whitney (*U)* test is an alternative to the *t*-test that does not require the data to follow a normal distribution. Unlike the *t*-test, which compares the mean values of two groups, the Mann-Whitney (*U)* test compares their medians. It examines the differences between two independent groups on a continuous scale.

With the aim of eliciting the reasons for the students' level of performance and statistical results, we also conducted an in-depth analysis aimed at detecting how the feedback generated by assessing the correctness of the programs triggered the reflections of the teacher and students, and resulted in new chains of actions aiming to attain better learning outcomes. We elaborated on two questionnaires, one focusing on the students, and the other focusing on the teacher, with the intent to generate qualitative data for an in-depth analysis.

Participants and procedures

This study was carried out during introductory programming courses, which were offered to the students during the first semester. None of the students had previous experience with programming when they started the course. Each semester, there are 40 students entering the computer science program. For the introductory programing courses, the students are always subdivided into two groups. This division allows the teacher to provide greater attention to the students. In some cases, the cohorts exceeded the expected number of 20 students because some of the students were repeating the course.

The programming language, which was used between 2011 and 2014, was the C programming language, and the course structure covered elementary data types, simple and structured commands, arrays, and matrices. The choice of C as the programming language was dictated by the pedagogical requirements of the course. Because we strove to maintain the variations in the variables to a minimum, using a single programming language allowed for a more consistent environment for comparing the assessment approaches. However, we do not believe that the choice of a different programming language would have had a major implication regarding the results because the feedback given during the assessment was basically whether the solution is correct or not. Any differences would have been due to the characteristics of the language and not to the assessment approach.

The study, conducted from 2011 to 2014, comprised 136 undergraduate students, and involved six cohorts and one teacher. The notation used here is YYYY/S, where the first four digits correspond to the year (YYYY), followed by the semester (S). For example, 2012/2 stands for cohort 2012 and the second semester.

Table 1 presents the analyzed cohorts, showing the number of students in each cohort (*N*) and their median final grade. In addition, the table shows which cohort used a continuous assessment (CA), and in which evaluation tasks the BOCA system was used. Depending on the cohort, the BOCA system was applied for CA tasks, exams, or exercises. Some cohorts had a CA but did not use the BOCA system for checking their correctness. The CA tasks were programming exercises that the students solved using a pencil and paper in class. The grades for the cohorts with no CA were composed of three exams taken during the semester. The grades for cohorts with a CA were made up of three exams and several small quizzes taken each week. Each student conducted all of the activities. For all of the cohorts, the students received only feedback regarding the correctness of their program.

In classes where the BOCA system was used, there were some particularities in the number and variety of activities undertaken with the BOCA system. For cohort 2011/2, all nine continuous assessment tasks were carried out using the BOCA system; six, using an automatic response; and three, with the automatic response option disabled. During all other activities (exercise lists, classroom exercises, etc.) the BOCA system was not applied. Cohort 2012/1 had an average of six continuous assessment tasks that did not use the BOCA system, and three exams that did. All other activities were carried out without the BOCA system. For cohorts 2012/2, 2013/2, 2014/1, and 2014/2, BOCA was used during all activities. Extra class activities were left on an online BOCA server, allowing the students to do them from home. In addition, at the beginning of the semester, a list of 40 exercises was left on the online BOCA server for students to do during the semester.

Table 1. Cohorts

Cohort	*N*	Median	CA	BOCA		
				CA	Exams	Exercises
2011/2	25	2.44	Yes	Yes	Yes	No
2012/1	29	5.52	Yes	No	Yes	No
2012/2	19	3.65	Yes	Yes	Yes	Yes
2013/2	23	1.92	No	-	Yes	Yes
2014/1	18	2.58	No	-	Yes	Yes
2014/2	22	2.82	No	-	Yes	Yes

For all of the cohorts, the programming activities assigned to the students were extracted from the Brazilian Informatics Olympiad (Anido & Menderico, 2007). These exercises involved real-world problems whose solutions are computer programs. Within the problem definition, strict rules for the input and output data formats were established. Such strict rules are important for two reasons.

First, to specify a problem's solution (computer program), the students must abstract the data and requirements that are often not fully explicit in the problem definition. This makes the students apply their abstraction capability. Second, the student's programs must strictly obey the input and output rules defined in the exercise, allowing the BOCA system to evaluate the programs properly.

Learning outcome measurement of correctness

The learning outcome assessment of correctness addressed the following measurement criteria from 2011 to 2014:

- The grades varied from 0 to 10.0, and students needed to score at least 5.0 to pass.
- When a CA was applied, the final grade was given by the following formula: (average grade in ongoing tests + average grade in final exams)/2.

- When a CA was not applied, the final grade was given by the following formula: average grade of the final exams.
- Considering the application of a CA when using or not using the BOCA system, a program that correctly processed the datasets was considered totally correct and earned the maximum grade; otherwise, the program earned a zero. The maximum grade for an ongoing test is 10.0, i.e., the sum of all programs is 10.0;
- Regarding the final exams, each exam contained four programs. The maximum grade for each program was 2.5. A program that correctly processed the datasets for multiple inputs earned a grade of 1.5. A program that reacted properly to an erroneous input earned an additional 0.5. A program that was compiled and run without errors earned another 0.5.

Questionnaires

The questionnaires were elaborated upon to extract qualitative data originating from the teacher's and the students' user experiences, new courses of action, and reflections on the content, knowledge, skills, abilities, and attitudes. The data collected from the questionnaires inform us the general level of satisfaction with the usage of the BOCA system, for example, whether the teacher found the BOCA feedback useful for improving their teaching, and if the BOCA feedback allowed immediate students or the teacher to make adjustments in response to student difficulties. Both questionnaires are described as follows.

Student questionnaire

1. Did the use of BOCA entice you to develop new forms of studying?
2. Did the use of BOCA encourage you to assess your skill levels?
3. Did the use of BOCA help you to detect your weaknesses?
4. Did the use of BOCA trigger any reflections on your performance?
5. How did you feel when using BOCA?
6. Did the use of BOCA motivate you to seek new content?
7. Did the use of BOCA motivate you to solve more exercises?
8. Did you like using BOCA?
9. Do you think you learned better with BOCA?
10. Did the use of BOCA generate group discussions?
11. Did you discover new things when using BOCA?
12. Do you think that BOCA meets your particular method of learning?
13. Did you overcome obstacles using BOCA?
14. Did the use of BOCA help in your development?
15. Did the use of BOCA lead you to try harder in subsequent tasks?
16. Did the use of BOCA bring about actions to improve your performance?
17. Was the use of BOCA a pleasant experience?
18. What was the overall balance from the use of BOCA?
19. Did the use of BOCA lead you to seek help from your teacher or classmates?
20. Did the use of BOCA involve you to engage in collaborative work with your classmates?

Teacher questionnaire

1. Did the use of BOCA lead you to develop new forms of feedback for students?
2. Did the use of BOCA help you to detect student weaknesses?
3. Did the use of BOCA trigger reflections on the performance of the students?
4. Did the use of BOCA motivate you to deliver new content?
5. Did the use of BOCA motivate to you deliver more exercises?
6. Do you think that you taught better when using BOCA?
7. Did the use of BOCA motivate you to hold discussions with the students?
8. Did you discover new things when you used BOCA?
9. Do you feel that BOCA meets your method of teaching?
10. Did you overcome any educational obstacles using BOCA?
11. Did the use of BOCA help you in your teaching?
12. Did the use of BOCA lead you to try harder in subsequent teaching activities?
13. Did the use of BOCA result in actions to improve your teaching?

14. Was the use of BOCA a positive experience?
15. What was the overall balance from using BOCA?
16. Are the data generated from BOCA sufficient to formulate effective teacher feedback to the students?

Empirical results

To answer the research questions of this work, we conducted a study in which we interpolated qualitative and quantitative techniques to analyze the data obtained from the students' grades, as well as data that came from the students and teacher questionnaires.

Statistical results

Considering those cohorts that used and did not use the BOCA system, and those cohorts that used and did not use a CA, the median distribution of the final grades are visualized in Figure 1. As shown in Figure 1, there was a slight difference in grades, indicating better results in classes that used a CA and in classes where the Boca system was not used. However, it is necessary to check whether these differences are statistically significant.

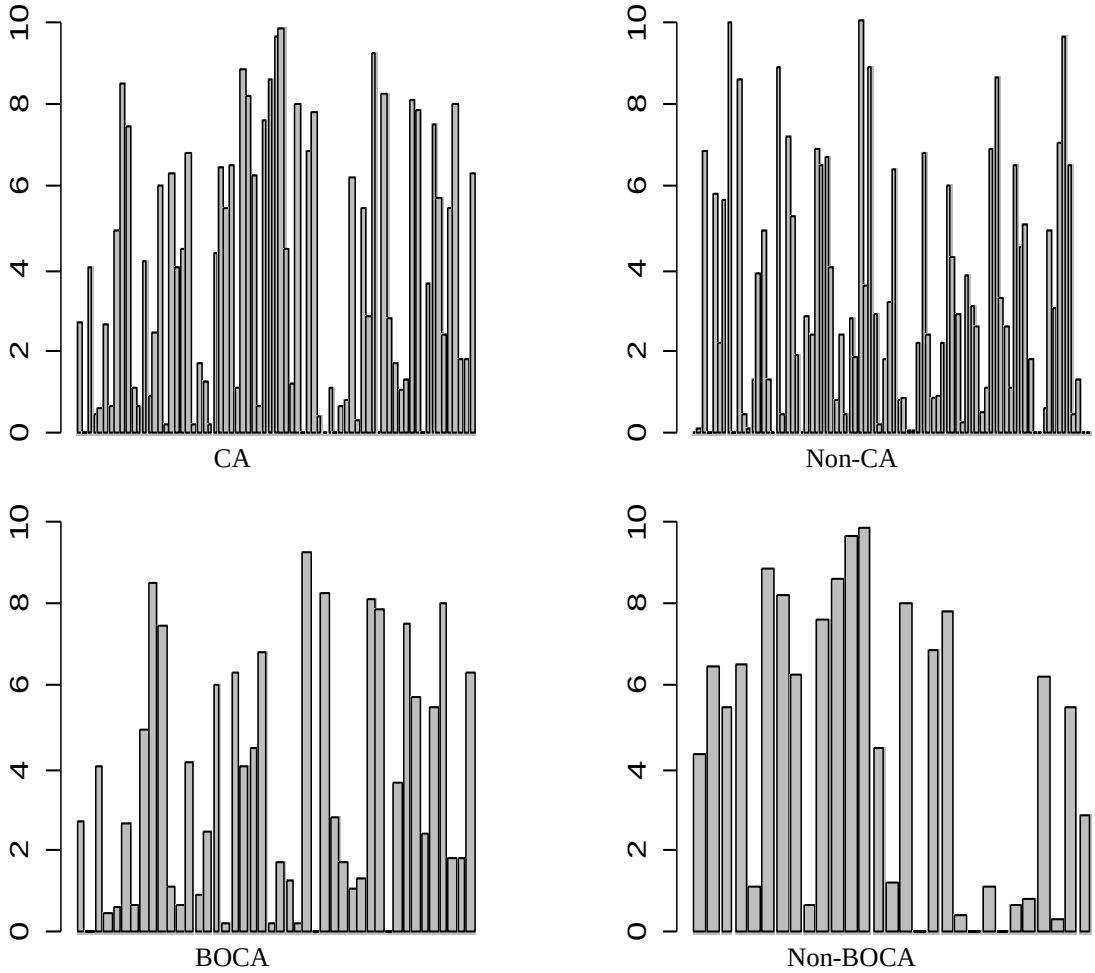

Figure 1. Visualized median of the final grades when applying/not applying a CA and the BOCA system

The grade distribution was analyzed by means of the Mann-Whitney (*U*) test. The Mann-Whitney (*U*) test was conducted in order to answer the first research question (RQ1), considering null hypothesis H_0: there is no significant difference between the median of the final grades of the cohorts that used a CA (2011/2, 2012/1, and 2012/2) and those that did not (2013/2, 2014/1, and 2014/2). The pertinent test is H_0: $\mu_X = \mu_Y \times H_1$: $\mu_X \neq \mu_Y$. Here, μ_X and μ_Y are the medians of cohorts X and Y, respectively, where X is the group of students that used a CA, i.e., cohorts 2011/2, 2012/1, and 2012/2. In addition, Y is the group of students that did not use a CA, i.e., cohorts 2013/2, 2014/1, and 2014/2. When comparing the grades of the cohorts, the result of the statistic test *W*

was 3331.5 and the *p*-value was 0.22, indicating a level of 5%, which means that, for the majority of the cohorts studied, no significant difference was found between the scores of those who used a CA and those that did not (Figure 2).

Additionally, the Mann-Whitney (*U)* test was conducted in order to answer the second research question (RQ2), with null hypothesis H_0: there is no significant difference between the median of the final grades of the cohorts that used the BOCA system (2011/2 and 2012/2) and those that did not (2012/1). The settings for these assumptions are the same as previously described.

The Mann-Whitney (*U)* test showed that there is no significant difference in the distribution of grades among the cohorts' instances, indicating that the feedback from a CA regarding the correctness of a program given to the students when using or not using the BOCA system did not help the students to increase their grades.

Figure 3 shows QQPlots for the previous configuration. Observed at a level of 5%, we can conclude that there is no significant difference between the median scores of X (where BOCA was used) and Y (where BOCA was not used). The result of the statistic test *W* is 551.5 and the *p*-value is 0.33.

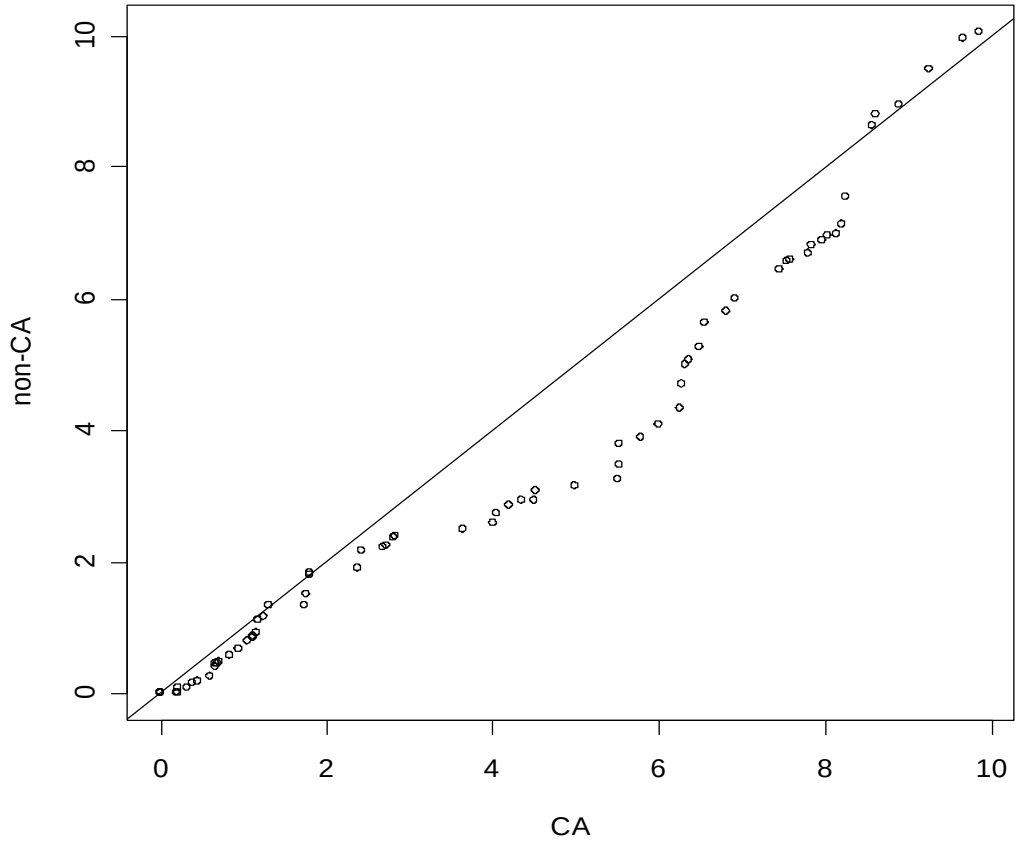

Figure 2. QQPlot comparing the average scores obtained for cohorts where a CA was used, and for cohorts where a CA was not used

Figures 2 and 3 show QQPlot graphs for situations related to the research questions. QQPlot graphs are used to compare the characteristics between two populations. QQPlot graphs contain dots representing the quantiles of each sample. If two samples come from the same population, then the dots should be along the 45º diagonal line starting at the origin. The samples can be compared based on their distribution, checking whether the points plotted on the graph are distributed near the 45º diagonal line. When the points plotted on the graph are parallel to the diagonal line, the two distributions are similar. We adopted the conventional a = 5% probability to determine a statistical significance.

An interpretation of Figures 2 and 3 corroborates the idea that the use of a CA when applying or not applying the BOCA system was not more effective, but when a CA was applied, the results were slightly better. In both situations, the points representing the grades are distributed near the diagonal line. We must mention here that in both situations where the BOCA system was used, the teacher was applying a CA. However, there were differences regarding the intensity of the feedback and the method used in correcting the programs. The

assessment of correctness was automatic, and the student feedback regarding the program correctness was given more frequently.

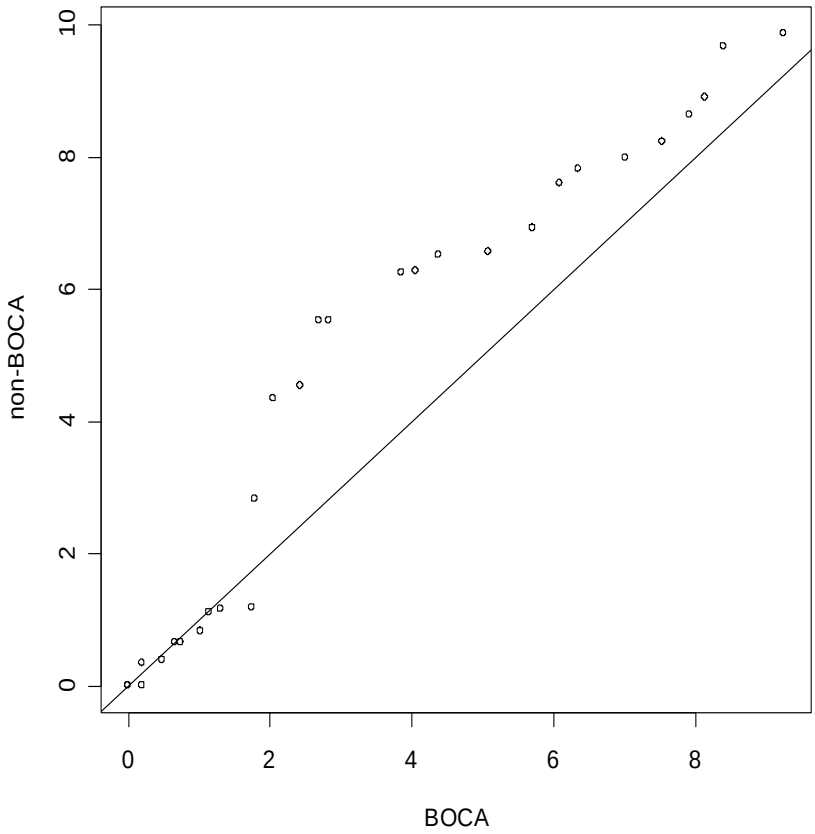

Figure 3. QQPlot comparing the grades of cohorts that used BOCA and the cohorts that did not. In all cohorts a CA was performed

It is worth mentioning that, as shown in Table 1, the median was 5.52 for cohort 2012/1, which is much higher than those of the remaining cohorts. Considering that for cohort 2012/1, the Boca system was not used for either a CA or the exercises, we can state that this is evidence reinforcing the results.

We expected that the continuous feedback regarding the level of correctness was not going to improve the students' grades much, given its limitations. However, we did expect that automatic and more frequent feedback of correctness could boost the students' performance, even to a small degree. To determine the reasons for the apparent inconsistency in the results, we delivered a questionnaire to the students and teacher focusing on whether the BOCA system could trigger the students' and teacher's reflections and actions, and if so, to what extent.

In-depth analysis

For a qualitative analysis, an in-depth analysis was conducted on the data collected from the questionnaires with the aim of enlightening the statistical results. The in-depth analysis covered the students' and the teacher's reflections, actions, and user experience.

Regarding the students' reflections, 82.6% of the students affirmed that the use of the BOCA system encouraged them to assess their skills, 73.9% said that the use of BOCA helped them detect their weaknesses, 67% answered that they learned better with BOCA, 42% discovered new things when using BOCA, 63.3% responded that BOCA meets their method of learning, 73.9% overcame obstacles using BOCA, 76.5% replied that the use of BOCA helped in their development, and 64.2% thought that their performance was improved. These results indicate that the students' perceptions regarding their improvement in knowledge were positive. In addition, all of the answers were positive, indicating that the use of BOCA led them to reflect on their failures.

Concerning the students' actions, 69.8% developed new forms of study using BOCA, and 70.3% answered that the use of BOCA led them to try harder in subsequent tasks. These facts indicate that BOCA lead the students to embrace a reactive attitude with the objective of overcoming their difficulties. This positive outcome, which is related to the students' actions along with the fact that BOCA also boosted their reflections, assures us that BOCA had a positive impact in student learning, and has potential to trigger student reflections and actions. Thus, we can conclude that the cause for the lower grades of students in cohorts in which the BOCA system was applied was not the absence of student reflections or corrective actions. These results corroborate the importance of providing feedback regarding the correctness to the students, showing that it has the potential to boost the students in making forward steps toward a deeper level of learning.

A possible explanation for the decreased performance of the students when using BOCA was the lack of collaborative and interactive practices. Only 30.9% of the students stated that the use of BOCA generated group discussions, 34.1% replied that the use of BOCA led them to seek help from their teacher or classmates, and 32.9% responded that the use of BOCA induced them to be involved in collaborative work with their classmates. Based on this, we have evidence showing that students had a bad social experience. They took individual actions to overcome obstacles, but were not motivated to work in groups when using the BOCA system. The use of the system could have facilitated a more individualized study, in detriment to collaborative learning and interactions with the teacher.

In relation to the students' experience during the use of the BOCA system, 63.2% felt enthusiastic when using BOCA, 50.8% affirmed that the use of BOCA motivated them to seek new content, 63.8% responded that the use of BOCA motivated them to solve more exercises, 59.2% liked using BOCA, 60.9% had a pleasant experience while using BOCA, and 80.0% considered the overall balance when using BOCA to be a good experience. These results indicate a positive affect concerning the use of BOCA. Most of the students enjoyed themselves while programming when using BOCA. In addition, they were motivated to use the system. We can therefore discard a lack of motivation as a possible cause for the lower performance of those students that used the BOCA system.

Students included additional comments highlighting some of the shortcomings related to the BOCA system's inability to show what actually is wrong in their learning when a program is incorrect. These observations reinforce our previous argumentation concerning the limitations of simply providing feedback regarding the correctness of a program, in which only the program is evaluated, but not the learning processes involved during the act of programming. In addition, a lack of student interactions and collaborations could have been the cause for the students' poor performance when using BOCA. These results are in line with what was previously addressed in this work. We deemed the attention given to student interactions and collaborations as essential to learning how to program. Some student excerpts are as follows:

> *"BOCA works well for programming Olympiads, but not for learning, where knowing what really went wrong helps a lot."*
>
> *"BOCA does not show how we fail."*
>
> *"BOCA does not evaluate the learning, it only evaluates the goal."*

With respect to the teacher reflections, the teacher responded that the use of BOCA triggered reflections on the performance of the students, but the teacher did not discover new things when using BOCA.

Regarding the teacher actions, the teacher answered that the use of BOCA lead him to develop a few new form of feedback for his students, and that it helped him to detect student weaknesses. However, he was not able to overcome educational obstacles using BOCA, and the use of BOCA slightly helped him in teaching, led him to try slightly harder in subsequent teaching activities, and resulted in few actions improving his teaching. In addition, the data generated from BOCA did not provide sufficient information to formulate effective teacher feedback to the students.

Concerning the teacher's experience, the teacher replied that the use of BOCA did not motivate him to deliver new content, but did motivate him to provide more exercises. In addition, he considered that he taught better with BOCA on only a few occasions, and that the use of BOCA motivated him to have greater discussions with the students, but also on only a few occasions. He considered that BOCA meets his method of teaching and was a positive experience, and he considered the overall balance from the use of BOCA to be good.

In general, the teacher's responses indicate that the use of BOCA neither led him to make an effort to try and teach better, nor provided sufficient information to understand the students' limitations. However, he considered the use of BOCA to be a good experience and a practice that meets his method of teaching. This is evidence that the teacher is accommodated and does not seek to become a better teacher. The results from the teacher questionnaire lines up with the idea that teachers must be warned regarding the limitations of only assessing the correctness of a program, and that they should be motivated to seek new interactive methods for assessing the students in order to unravel the educational problems that lead the students to build an incorrect program.

Conclusions

The objective of this work was to motivate teachers to develop more interactive teaching and learning practices, establish links between empirical results and key issues regarding programming feedback, and going slightly further, address the limitations and potentialities of using an assessment of program correctness as a vehicle for improving student learning.

Empirical research conducted from 2011 to 2014 in an introductory programming course gave us insight into common CA practices, indicating that there is a need for better evaluation processes for determining student performance and ways for dealing with relevant assessment information. This research highlighted the need for resources and practices that go further than simply applying ongoing tests, and providing only feedback for a program's correctness when using or not using the BOCA system.

Empirical results show that there is no significant difference between the students' performance when comparing the cohorts that used and did not use the BOCA system and a CA, indicating that a CA using BOCA with only automatic feedback for correctness, or applying a CA where the teacher simply checks for correctness, does not significantly improve the teaching and learning processes of computer programming. The Mann–Whitney test did not indicate a difference at a 5% level between the cohorts that used and did not use BOCA, and between the cohorts that used and did not use a CA. Although this difference is visible in our dataset, the result is not statistically significant.

Automatic program correction systems are widely used in programming marathons or Olympiads worldwide. This is due mainly to the large number of submissions to be assessed within a short period of time. Certainly, the use of systems such as BOCA assist in giving feedback more quickly, and in undergraduate programming disciplines, quick feedback to students is extremely important. However, the quality of the feedback should also be taken into account, and BOCA feedback is necessary for triggering student reflections and corrective actions, but is limited and insufficient to imply a significant improvement in students. Thus, we believe that the use of BOCA is important, but should be accompanied by interactive teaching strategies and the teacher's facilitation of collaboration among the students.

Another issue of discussion is whether the use of a CA regarding program correctness helps teachers and students to correct any mistakes early in the teaching/learning processes through more knowledge evaluation checkpoints. A large number of assessments and quick feedback on student performance triggered a slight amelioration in student performance. However, this was proven to be an inefficient practice regarding the teaching and the students' understanding of the reasons that lead to an incorrect program.

We suggest that diversified teacher/student interactions using BOCA can aid in student achievement, as long as automatic feedback from BOCA is used sparingly. Given that automatic feedback basically indicates whether a solution is right or wrong, a student who submits a program with an error, even if the program contains only small mistakes, will not find any support helping to fix the student's program. Thus, teachers must interact with their students to be aware of which problems cause programming mistakes, and provide quick feedback with better quality than BOCA can offer.

In conclusion, the use of BOCA and the application of a CA for verifying the correctness of a program correspond to a minimally informative practice. The use of BOCA and the application of a CA regarding only program correctness are a necessary but insufficient assessment method. Noteworthy efforts are required to improve the use of a CA in practice.

The aim of this work was to foster teachers to go beyond simply checking for the correctness of their students' computational solutions. The adoption of assessment practices and environments that build a cumulative knowledge through multiple means of assessment is required, allowing valuable interactions among students,

and between the students and their teacher, regarding the students' solutions. Teaching reflection needs to be more interactive, investigative, and integrated into the students' individual methods of learning. There is a necessity for the teacher to develop interactive and collaborative strategies to reveal to the teachers and students data related to chains of cognitive breakdowns that lead to programming errors.

References

Anido, R. O., & Menderico, R. M. (2007). Brazilian Olympiad in informatics. *Olympiads in Informatics*, *1*, 5–14. Retrieved from http://www.ioinformatics.org/oi/pdf/INFOL014.pdf

Ashour, O. M., Sangelkar, S., Warley, R. L., & Oniped, O. (2014). Redesign the engineering teaching and assessment methods to provide more information to improve students' learning. In *Frontiers in Education Conference (FIE)* (pp. 1-6). doi:10.1109/FIE.2014.7044286

Brown, G. T. L., & Hattie, J. A. (2012). The Benefits of regular standardized assessment in childhood education: Guiding improved instruction and learning. In S. Suggate & E. Reese (Eds.), *Contemporary Debates in Child Development and Education* (pp. 287-292). London, UK: Routledge.

Bornat, R. (2011, March). *Some problems of teaching (learning) first-year programming (plus a glimmer of hope)*. Paper presented at Information and Computer Science: 11th Programming Workshop. Retrieved from www.researchgate.net/publication/266342475

Carter, J., English, J., Ala-Mutka, K., Dick, M., Fone, W., Fuller, U., & Sheard, J. (2010). How shall we assess this? *SIGCSE Bulletin*, *35*(4), 107-123. doi:10.1145/960492.960539

de Campos C. P., & Ferreira C. E. (2004, August). *BOCA: um sistema de apoio a competições de programação* [BOCA: A Support system for programming contests]. Paper presented at Brazilian Workshop on Education in Computing, Salvador, Brazil.

Denadhi, S. (2009). *A Cognitive study of learning to program in introductory programming courses* (Unpublished doctoral dissertation). Middlesex University, London, UK.

Dijkstra, E. W. (1989). On the cruelty of really teaching computing science. *Communications of the ACM, 32*(12), 1398-1404.

Earl, K., & Ussher, B. (2012). Confidence in assessment decisions when using ICT. *Computers in New Zealand Scholls*, *24*, 90-107.

Fisher, D., & Frey, N. (2007). *Checking for understanding: Continuous assessments for your classroom*. Alexandria, VA: ASCD.

Hattie, J., & Timperley, H. (2007). The Power of feedback. *Review of Educational Research*, *77*, 81-112.

Ihantola, P., Karavirta, V., & Seppälä, O. (2011). Automated visual feedback from programming assignments. In *Proceedings of the 6th Program Visualization Workshop* (PVW'11) (pp. 87–95). Retrieved from https://research.aalto.fi/files/4370307/VisualFeedback.pdf

Jenkins, T. (2002). On the difficulty of learning to program. In *Proceedings for the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences* (pp. 53-58). Retrieved from http://www.psy.gla.ac.uk/~steve/localed/jenkins.html

Jerinić, L., Ivanović, M., Putnik, Z., Budimac, Z., & Savić, M. (2014). e-Education in teaching programming – Forty years of promises? In *Proceedings of International Conference on e-Learning* (Vol. 14, pp. 225-233). Retrieved from http://elearning-conf.eu/docs/cp14/paper-34.pdf

Koulouri, T., Lauria, S., & Macredie, R. (2014). Teaching introductory programming: A Quantitative evaluation of different approaches. *ACM Transactions on Computing Education*, *14*(4), 26. doi:10.1145/2662412

McCracken, M., Almstrum, V., Diaz, D., Guzdial, M., Hagan, D., Kolikant, Y. B. D., Laxer, C., Thomas, L., Utting, I., & Wilusz, T. (2001). A Multi-national, multi-institutional study of assessment of programming skills of first-year CS students. *ACM SIGCSE Bulletin*, *33*, 125-180. doi:10.1145/572139.572181

Messick, S. (1989). Validity. In R. Linn (Ed.), *Educational Measurement* (3rd ed., pp. 13–103). Washington, DC: American Council on Education Macmillan.

Muller, O., Haberman, B., & Averbuch, H. (2004). (An almost) pedagogical pattern for pattern-based problem-solving instruction. In *ACM SIGCSE Bulletin* (Vol. 36, No. 3, pp. 102-106). New York, NY: ACM.

Nicol, D. J., & Macfarlane-Dick, D. (2006). Continuous assessment and self-regulated learning: A Model and seven principles of good feedback practice. *Studies in Higher Education*, *31*(2), 199-218.

Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A Review and discussion. *Computer Science Education, 13*(2), 137-172.

Scott, M. J., Counsell, S., Lauria, S., Swift, S., Tucker, A., Shepperd, M., & Ghinea, G. (2015). Enhancing practice and achievement in introductory programming with a robot Olympics. *IEEE Transactions on Education, 58*(4), 249-254.

Scriven, M. (1967). The Methodology of evaluation. In R. W. Tyler, R. M. Gagne, & M. Scriven (Eds.), *Perspectives of Curriculum Evaluation* (pp. 39-83). Chicago, IL: Rand McNally.

Serra-Toro, C., Traver, V. J., & Amengual, J. C. (2014). Promoting student commitment and responsibility through self- and peer-assessment. In *Frontiers in Education Conference* (pp. 1811-1814). doi:10.1109/FIE.2014.7044285

Soloway, E., & Ehrlich, K. (1984). Empirical studies of programming knowledge. *IEEE Transactions on Software Engineering*, *10*(5), 595-609.

Spohrer, J. C., & Soloway, E. (1986). Novice mistakes: Are the folk wisdoms correct? *Communications of ACM*, *29*(7), 624-632.

Tang, Y. (2009). To develop the students' creativity in the lecture of C programming. In *First Educational Workshop on Education Technology and Computer Science* (pp. 790-794). doi:10.1109/ETCS.2009.712

Vihavainen, A., Airaksinen, J., & Watson, C. (2014). A Systematic review of approaches for teaching introductory programming and their influence on success. In *Proceedings of the 10th International Computing Education Research Conference* (pp. 19-26). New York, NY: ACM.

Utting, I., Tew, A. E., McCracken, M., Thomas, L., Bouvier, D., Frye, R., Paterson, J., Caspersen, M, Kolikant, Y. B. M., Sorva, J., & Wilusz, T. (2013). A Fresh look at novice programmers' performance and their teachers' expectations. In *Proceedings of the ITiCSE working group reports conference on Innovation and technology in computer science education-working group reports* (pp. 15-32). New York, NY: ACM.