JOURNAL
OF EDUCATIONAL SCIENCES & PSYCHOLOGY
Vol. I (LXIII)
No. 1/2011
51 - 56

Learning behaviors measured by a web-based self-assessment system

Margit Antal[a*], Szabolcs Koncz[b]

[a]*Sapientia University, Tirgu Mures, Romania*
[b]*Petru Maior University, Tirgu Mures, Romania*

Abstract

In this paper, we present an empirical study related to the use of a web-based self-assessment system in an object-oriented programming course. 44 science and engineering students participated in this study, which aims to investigate the following questions: (1) How much time do students spend voluntarily on using the self-assessment system? (2) What kind of tests do students prefer? (3) What is the relation between theoretical knowledge and problem solving skills? Results show an increase in the overall learning outcome, moreover the analysis of the data helped to reveal a group of students with learning difficulties.

Keywords: computer based testing, web-based learning, self-assessment

1. Introduction

Assessment is used to evaluate an educational program and according to the literature, it can be formative or summative (Dunn, 2009). Summative or final assessment can be considered as an "assessment of learning", while formative assessment means monitoring student progress during the learning unit, therefore it is an "assessment for learning" (Chappuis & Stiggins, 2002). In this context, our paper evaluates both types of assessment. Moreover, it presents an evaluation of how formative assessment can be used for summative assessment.

Self-assessment systems (SAS) are usually parts of intelligent learning environments (ILE), which have a long history. ILE itself is considered as a generalization of intelligent tutoring system (ITS). A great deal of research papers present ILE systems and evaluate such systems as a whole (e.g., Huang et. al., 2009; Galvez et. al., 2009), which do not always provide enough information about special components of such systems. Therefore, Brusilovsky et. al. (2004) proposed a layered evaluation of these types of systems.

Considerable amount of research (e.g., Martinez & Martinez, 1992; Black & Wiliam, 1998) has been focused on the positive effects of formative assessment. Among these, there are very few studies focusing on the effects of self-assessment in the teaching-learning process. However, according to Ross (2006), a large proportion of teachers reports using it as an assessment method, which also demonstrates the validity and utility of this method in education.

For computer science graduates the most important ability is to produce code and solve programming problems. A number of automated program assessment systems have been produced until now. These assignment-oriented systems assess students' problem solving

* Corresponding author. Tel.: +40-265-250-620; fax: +40- 265-206-211
E-mail address: manyi@ms.sapientia.ro

and code producing ability. Programming knowledge is strongly influenced by the ability to use programming constructs in context, which can be assessed by question-oriented systems. Question-oriented systems typically use single choice, multiple-choice or fill in the gap type questions.

Despite the fact that question-oriented systems are less popular, we consider that students knowledge of semantics as well as their ability to understand programs contribute to the development of their problem solving ability, which is the final goal of programming subjects. Thus, we constructed a web-based self-assessment system *Intelligent* (2009), which belongs to the question-oriented systems category.

Since our SAS is a question-based assessment system, in the following we review a few self-assessment systems of this category. Brusilovsky & Sosnovsky (2005) present an evaluation of a web-based self-assessment system used in computer science education. One interesting feature of their system is its ability to generate questions from parameterized question templates. A single question template is able to produce a very large number of questions. They used the system for Programming in C course, and reported an increase in knowledge of semantics and higher level programming. Guzman & Conejo (2005) and Guzman et. al. (2007) present SIETTE a web-based adaptive SAS based on Item Response Theory (Lord, 1953; Baker, 2001). The above-mentioned studies explore how self-assessment tests influence students' final assessment. A recent study (Alaoutinen & Smolander, 2010) presents a methodology to motivate students using a progress-monitoring SAS.

2. Methodology

2.1. Objective

This study sought to detect patterns in the learning habits of students based on how these are reflected in a SAS. Particularly we were interested in finding answers for the following questions: (1) How much time do students spend voluntarily on using the self-assessment system? (2) What kind of tests do students prefer? (3) What is the relation between theoretical knowledge and problem solving skills? Finally we analyze the overall learning outcome.

2.2. Participants

Our system was evaluated during October 2009 - February 2010 for an object-oriented programming course. Object-oriented programming is taught through the Java programming language. Although 57 engineering and science students participated in this study, only 44 passed the examination. Thus the data used to build the statistics comes only from those students who passed the exam.

2.3. Instrument

The *Intelligent* (2009) system is a web-based self-assessment system based on the classical test theory. The system is based on an item bank, which contains three types of questions: fill in, single choice and multiple choice. Each item belongs to a single topic and it is characterized by a difficulty parameter having one of the following values: “very easy”, “easy”, “medium”, “hard” and “very hard”. Each item can have an explanation attached to it, which is visible after the test is completed.

A major advantage of our system is the variety of test types offered, which can be further enriched. Currently, there are four test types in our test system: *standard*, *practice*, *objective-wise* and *custom*.

Standard tests are created by teachers and are similar to tests used in final assessments. Attention must be paid to their design in order to contain well-balanced items from all

topics. *Practice tests* contain items from all topics and can be used for practice before examinations. In this case students may select among elementary, intermediate and advanced tests. The items are generated randomly, respecting the criterion of difficulty (elementary tests contain only very easy and easy items, intermediate test contain easy and medium difficulty items and advanced tests contain hard and very hard items). In the case of *Objective-wise* test a topic is selected first from which the system generates a number of random items. The number of generated items is a parameter of the system. *Custom tests* are created by teachers, and in this case there are two options: (i) to select the topics and the difficulty of items to be generated in the test; items would be generated randomly; (ii) to select manually the items. Custom tests are very practical when the tutor wants to test the students on a limited number of topics.

2.4. Procedure

The final examination has two parts: a computer test using the self-assessment system, and a problem solving part. The final computer test was generated using an item bank. This consists of items grouped into 9 topics: basics of Java programming, classes and objects, arrays and strings, object-oriented programming, exception handling, nested classes, streams, threads, collections and maps. The test part of the final exam assessed the theoretical knowledge of these topics. The second part measures the problem solving skills of students using the object-oriented programming concepts. The following topics were considered mainly: simple class design (ex. Document class), class design using one-to-many relationships between classes (ex. Document store), inheritance (ex. Text document), collections and threads.

During the first lecture students were informed about the examination methods and the SAS has been available from the beginning. The usage of the SAS during the term was not compulsory. It was offered as a tool to facilitate students' preparation for the examination.

2.5. Data analysis

For data analysis we used R (The R project for Statistical Computing, n.d.), which is a free software environment for statistical computing. One of the strengths of R is that it includes an effective programming language and it can be easily integrated as a third party software. Although R offers a wide range of clustering approaches, Ward’s method (Ward, 1963) for hierarchical clustering produced the most meaningful clusters.

3. Results

In the following, we are going to answer the questions put in the introduction.

How much time do students spend on using the self-assessment system?

We wanted to know how regularly students practice self-assessment during the term. It turned out that 80% of the tests were solved in the examination period, and only 20% of the tests were solved during the term. This shows the fact that only a small group of students use the available educational tools without being forced. Every student used the SAS to prepare for the exam for on average of 4.3 hours.

What kind of tests do students prefer?

During the term, students were allowed to practice using the SAS. This resulted in 26,488 solved tests: 62.93% objective-wise tests and 37.07% practice tests (6.80% advanced 17.78% intermediate and 12.50% elementary). There is a clear preference for objective-wise tests, which means that students like to learn in a 'topic-by-topic' manner. Although the vast majority of our students chose practice tests after they had solved a few tests from every topic, there were a few students solving only practice tests.

What is the relation between theoretical knowledge and problem solving skills?

We clustered the final test marks and problem solving marks using the R statistical package. Ward's hierarchical clustering method was used, where the distances were computed using the Euclidean metric. The hierarchical cluster analysis identified four clusters shown in Figure 1. The corresponding statistical data is listed in Table 1.

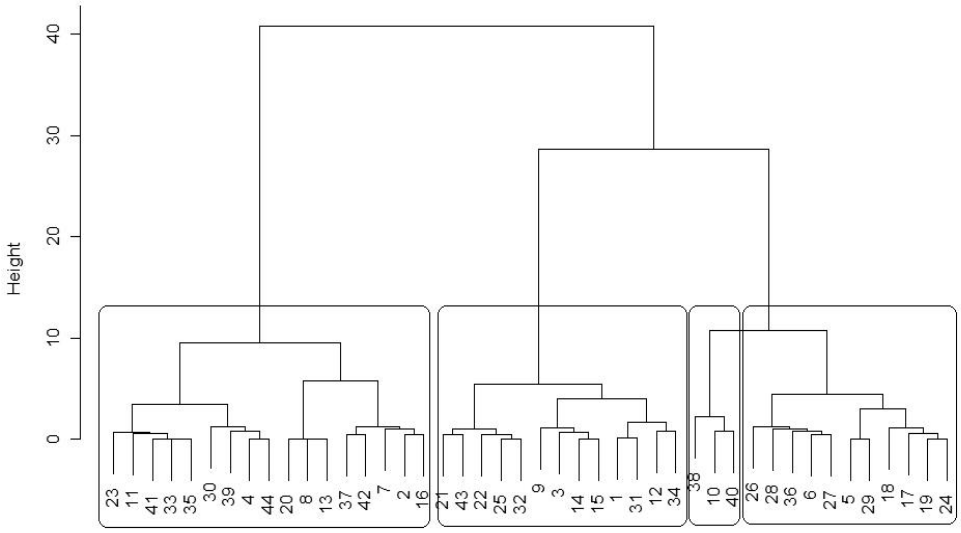

Figure 1. Clusters identified by a hierarchical clustering algorithm

Let us analyze these clusters going from left to right. The students inside the first cluster are the advanced students, whose theoretical knowledge level is almost identical with their problem solving skills level. The major part of this group is formed by students who had also learned programming subjects in the high school.

Table 1. Students clusters statistics

	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Number of students	17	13	3	11
Final test average	9.04	9.14	5.37	5.93
Problem solving average	8.89	5.58	8.67	5.50
Final test variance	0.66	0.56	0.76	0.57
Problem solving variance	0.92	0.53	0.77	0.41
Average time (minutes)	312	309	70	169

Students whose theoretical knowledge is much higher than their problem solving skills populate the second cluster. These are the studious students who like to learn theory, but they have problems with programming. Two thirds of these students started to learn programming at the university. These students were successful in learning the theoretical concepts, but the lack of basic programming skills prevented them from solving object oriented programming problems.

The third cluster is the smallest one, and students whose problem solving knowledge is much higher than their theoretical one populate it. Despite the fact that they had not spent

enough time to learn theory, they were able to solve problems using objects. These students like to learn exclusively by developing software.

4. Discussions

In conclusion, we can state that the self-assessment system brings many advantages to the students and helps their preparation for the final exam. The system helped various student categories differently. Some of the students used it mainly as a review tool the day before examination; other students used the tool daily in the exam preparation period. Those using the day before examination solved mainly practice tests (easy, intermediate and advanced), while the others solved mainly objective-wise tests. The usage of the self-assessment system was very instructive for the instructors too as it permitted to monitor the learning habits of students.

All the students appreciated the freedom of working from home and the freedom of choosing the test type. They also appreciated having access to their progress report, which raised the self-confidence level of less able students regarding the subject. We consider that the self-assessment system raised the transparency of the exam, too. Moreover, the feedbacks provided by the students will help us to develop further the system. We have already learnt that it is not enough to create good educational systems; we also have to teach our students how to use it regularly.

The main goal of the self-assessment system (SAS) was to help the students' in their preparation for the object-oriented programming examination and this was achieved. Moreover, the students' preference for objective-wise tests revealed the fact that the majority of the students prefer to learn theory in a 'topic-by-topic' manner. The analysis of traces left by students in the SAS permitted us to identify a group of students facing learning difficulties (the second cluster). These students learnt theory successfully but they were unable to apply the theory for problem solving. The analysis of exam results' revealed that this group of students had serious problems with basic programming skills, which prevented them in developing object-oriented programming skills.

This year we are using *Intelligent* in the same course. We have already finished the development of a computer adaptive test system based on Item Response Theory, which will be used for the final test. The following developments aim to make the system Question and Test Interoperability QTI (IMS Question & Test Interoperability Specification, n. d.) compatible, which is the de facto standard for computer based testing.

References

1. Alaoutinen, S., Smolander, K. (2010). Student Self-assessment in a Programming Course Using Bloom's Revised Taxonomy. *Proceedings of the fifteenth annual conference on Innovation and Technology in computer science education*, 155-159.
2. Baker, F. B. (2001). *The Basics of Item Response Theory* (2nd edition). ERIC Clearinghouse on Assessment and Evaluation, College Park, Maryland University.
3. Black, P., Wiliam, D. (1998). Assessment and classroom learning. *Assessment in education*, 5(1), 7-74.
4. Brusilovsky, P., Karagiannidis, C., Sampson, D. (2004). Layered evaluation of adaptive learning systems. *Int. J. Cont. Engineering Education and Lifelong Learning*, 14 (4/5), 402-421.
5. Brusilovsky, P., Sosnovsky, S. (2005). Individualized Exercises for Self-Assessment of Programming Knowledge: An Evaluation of QuizPACK. *ACM Journal of Educational Resources in Computing*, 5(2).
6. Chappuis, S., Stiggins, R. J. (2002). Classroom assessment for learning. *Educational Leadership*, 60(1), 40-43.

7. Dunn, K., E. (2009). A Critical Review of Research on Formative Assessment: The Limited Scientific Evidence of the Impact of Formative Assessment in Education. *Practical Assessment, Research & Evaluation*, 14(7), 1-11.
8. Galvez, J., Guzman, E., Conejo, R. (2009). A blended E-learning experience in a course of object oriented programming fundamentals. *Knowledge-Based Systems*, 22, 279-286.
9. Guzman, E., Conejo, R. (2005). Self-assessment in a feasible, adaptive web-based testing system. *IEEE Transactions on Education*, 48(4), 688-695.
10. Guzman, E., Conejo, R., Pérez-de-la-Cruz, J.-L. (2007). Improving Student Performance Using Self-Assessment Tests. *IEEE Intelligent Systems*, 22(4), 46-52.
11. Huang, Y.-M., Lin, Y.-T., Cheng, S.-C. (2009). An adaptive testing system for supporting versatile educational assessment. *Computers & Education*, 52, 53-67.
12. IMS Question & Test Interoperability Specification. (n. d.). URL http://www.imsglobal.org/question/
13. Intelligent. (2009). URL http://ms.sapientia.ro:8009/oop/
14. Lord, F. M. (1953). The relation of test score to the trait underlying the test. *Educational and Psychological Measurement*, 13, 517-548.
15. Martinez, J. G. R., Martinez, N. C. (1992). Re-examining repeated testing and teacher effects in remedial mathematics course. *British Journal of Educational Psychology*, 62, 356-363.
16. Ross, J., A. (2006). The Reliability, Validity, and Utility of Self-Assessment. *Practical Assessment, Research & Evaluation*, 11(10), 1-13.
17. The R project for Statistical Computing. (n. d.). Retrieved from: http://www.r-project.org/
18. Ward, J. H. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301), 236-244.