

A Method for Automated Program Code Testing

Sigitas DRASUTIS, Vida MOTEKAITYTĖ, Algirdas NOREIKA

Kaunas University of Technology, Faculty of Informatics

Studentų 50, LT-51368 Kaunas, Lithuania

e-mail: {sigitas.drasutis, vida.motekaityte, algirdas.noreika}@ktu.lt

Received: May 2010

Abstract. The Internet has recently encouraged the society to convert almost all its needs to electronic resources such as e-libraries, e-cultures, e-entertainment as well as e-learning, which has become a radical idea to increase the effectiveness of learning services in most schools, colleges and universities. E-learning can not be completely featured and met without e-testing. However, in many cases e-testing tools are suitable just for traditional/theoretical knowledge testing, covered by such items as questions, quizzes, matching boxes and other. The article “A Method for Automated Program Code Testing” tackles the lack of functions in e-testing systems and suggests e-assessment possibilities for students who study computer science, especially programming. The article analyzes the method that allows freely entering answers to questions, checking program syntax during the testing and enables automatic written code checking and evaluation.

Keywords: e-learning, distance learning, e-testing tools, e-testing method, e-testing environment, e-tests, computer adaptive testing (CAT), computerized testing, computer-based testing, automatic code verification, program code testing.

1. Introduction

The Internet and information communication technologies are more and more extensively involved in daily learning processes all over the world. This growth has been stimulated in part by the interest of educators and trainers in the use of new, internet based and multimedia technologies, as well as by the recognition that traditional ways of organizing the education need to be reinforced by innovative methods intending to realize the fundamental right of all people to learning. Electronic learning syllabus, discussion tools, video conferences and e-assessment tools seem to be trivial today and are placed in the centre of today’s learning environment. It is visible that distance and e-learning with various information systems and virtual learning environments have taken the first place within such processes. Therefore, the increasing importance of e-learning and the integration of information systems in all areas of learning force teachers to understand and implement IS (information systems) and IS specialists to start thinking about more efficient and precise mechanisms for such system design and implementation, the development of new tools not just for making learning material more accessible but also for creating e-assessment tools. The use of computerized adaptive testing (CAT; Wainer, 2000) has substantially increased since it was first formulated in the 1970s (Meijer *et al.*, 1999).

Computerized testing greatly increases the flexibility of test management (Gibbons *et al.*, 2008). When an examinee administers a test via the computer, the computer can update the evaluation of the examinee's ability after each item and then the ability to estimate can be used in the selection of subsequent items. With the right item bank and a high variance of examinee's ability e-test can be more efficient than a traditional paper-and-pencil test (Gierl *et al.*, 2008).

In most cases e-testing tools are suitable just for traditional/theoretical knowledge testing while providing such items as questions, quizzes, matching boxes and other (Tonidandel *et al.*, 2002). Nowadays suggested CATs are not applicable for all subjects and skills. Most CATs are based on an item-response theory model, yet the item response theory is not applicable to all skills and item types. While analyzing advantages and disadvantages of CATs, Rudner (1998) claims that CATs require careful item calibration. The item parameters used in a paper-and-pencil testing may not always be applied and suitable in a computer adaptive test. Computer science tests especially related with programming fall into that category as well. A paradox, however, can be observed that in this information-communication technology age where computer programs are well and rapidly developed, students who learn programming are still manually tested.

Thus, this article deals with e-testing method that allows to freely enter answers to questions, check program syntax during the testing and enables automatic written code checking and evaluation. The method will be used for students, studying computer science, to test their practical abilities of programming.

2. E-Testing Today

The migration of standardized tests from traditional paper-and-pencil administration to computerized adaptive administration offers many potential benefits, including increased measurement efficiency, immediate scoring, and more frequent administration dates (Harmes *et al.*, 2001).

Paper-and-pencil tests (traditional testing) are typically "fixed-item" tests in which the examinees answer the same questions within a given test booklet. Computerized testing offers a number of options for timing and formatting. Therefore, it has the potential to accommodate a wider range of item types. Tests are individually paced so that the examinee does not have to wait for others to finish before going on to the next section. Self-paced administration also offers extra time for the examinees that need it, potentially reducing one source of test anxiety (Georgiadou *et al.*, 2006; Bergstrom, 1992).

E-testing online assessments bring the best outcome through the reliable test engine which contains a test that reached the standards which went through the test creation process focusing on the reusability of the elements or big parts of tests (Brusilovsky *et al.*, 2002). E-testing not only reduces time but also plays a big role while helping the teacher to check students' knowledge and practical skills.

During the semester students practically work, studying algorithms and writing programs' code. Testing of student knowledge only with standard tests is not enough because

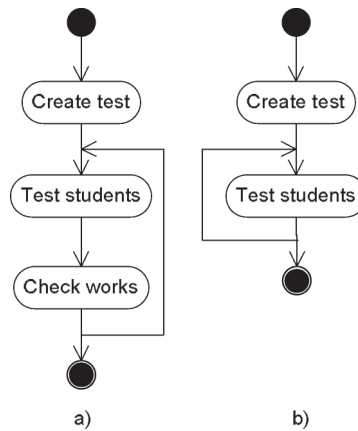


Fig. 1. A typical knowledge control process: a) a testing process without any tool; b) the e-testing process.

reading the question and understanding answers do not mean that student knows how to use his knowledge in a practical way (Tonidandel *et al.*, 2002). Therefore, most of the tests students are created by writing programs and checking them manually. A typical knowledge control process without and with e-testing environment from the lecturers' point of view is illustrated in Fig. 1 and includes several phases.

The difference seems to be minimal as the correction of students' works is eliminated, but in fact this task takes about 75% of all the time (Hardré *et al.*, 2006). But time saving is not the only advantage of e-testing system. E-testing tools present more useful functions: no manual evaluation is needed and statistics on question answering, student reports and many other things are provided.

E-test is one of assessment modes, administered on the computer, intended to measure the test-takers' or students' knowledge, skills and aptitudes. Knowledge is a set of theoretical information acquired on the subject and practical capability to apply that theoretical knowledge in learner's life (Brusilovsky *et al.*, 1999).

Existing learning management systems (LMS), suitable for e-learning, such as Blackboard Learning System Vista, Moodle, ATutor and other offer a wide range of tests made of questionnaires, quizzes, crosswords, multiple-choices, free responses and etc. But most of them propose tests and quizzes made just from questions and few answers (Fig. 2a).

Testing types, mentioned above, have become a popular way to assess the learner, check his/her knowledge; however, it has some disadvantages as well:

- All possible answers both correct and incorrect are visible which allows students to guess the right answer even if he/she does not know that subject at all.
- This type of knowledge assessment checks only the understanding of information (theoretical knowledge assessment) and capability to read and think. The practical usage of theoretical information, which is a very important aspect, remains untouched.

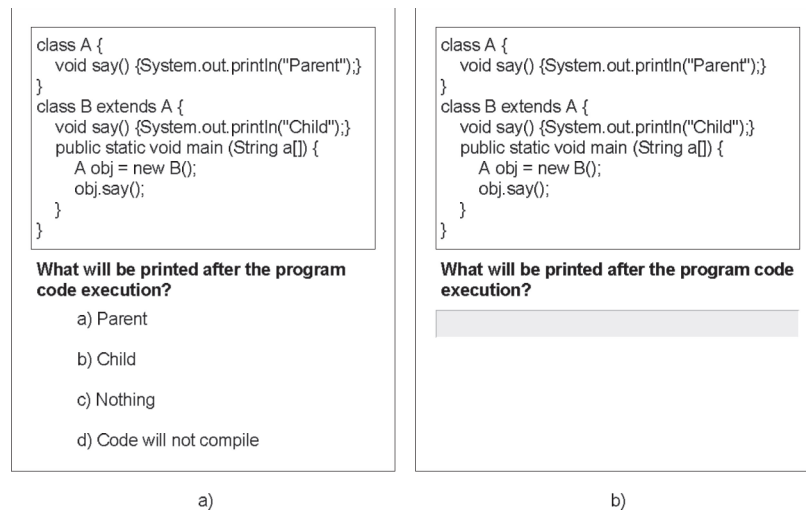


Fig. 2. E-test types: a) question with answers, b) question with free entered answer.

Some e-test types suggest possibility for students to enter free responses to questions (Fig. 2b). These answers are automatically compared by computer in straight equivalence with the right ones:

- This kind of testing is a better way to check knowledge than allow choosing answers from given possibilities because answers are invisible and students need to think more.
- Formulation of the answer depends on students' ability to express his/her thoughts, the understanding of the subject and the topic; therefore, the comparison one by one does not function properly.

Both ways leave the practical aspect away. But when speaking about computer science and information technologies, especially programming, it is obvious that testing of practical knowledge is essential. So there is a need for automated tests and program testing modules that help to reduce and ease the amount manual work, carried out by lecturers.

A lot of work has already been done by helping people to study programming languages and not only by reading books and finding information and examples over the Internet. W3Schools offer different materials on studying programming technologies. It offers Try-It-Yourself demos where it is possible to read the chapter and later try to change standard code examples for some programming languages like HTML¹, SQL², CSS³. It is very useful for students and it was great, if such experience was transferred into the world of tests.

¹HTML Examples. URL: http://www.w3schools.com/html/html_examples.asp.

²SQL Try It. URL: http://www.w3schools.com/sql/sql_tryit.asp.

³CSS Examples. URL: http://www.w3schools.com/Css/css_examples.asp.

3. Designing E-Testing Method

Carrying out the research and teaching students who study computer science it was observed that tests where students can enter the program code and the code is automatically checked could be very attractive and useful, because fast answering is possible as well as the reduction of teacher's work. This also allows enabling automatic checking of the code syntax's errors and provides opportunities to find and correct code errors because they are not essential and concentrate on algorithms.

The strongest part of such tests is a possibility to write program code in personal student style and check just the correctness of the result but not the way it is coded. This provides test's flexibility. However, this is only possible with programming languages where coded functions can output some results, like SQL query results, or C, JAVA, PHP function's returning values. Then the way the program code is written is not important as the emphasis is placed on the fact if the function provides correct answers with the test data.

Such tests face the problems indicated below:

- Result checking may determine wrong program code writing, using forbidden techniques or hard-coding the results.
- It is very hard to check the correctness of the program when the program code does not generate the exact output. For instance, checking written HTML code needs sophisticated code analysis which will not be discussed in this article.

This article tackles only the first problem indicated, which can be solved by entering allowable and disallowable keywords for the entered program code and checking answers not for syntax correctness but also for using such keywords.

The second problem will not be analyzed in this article because the main purpose of the paper seeks to concentrate on programming languages that maintain functions which return exact results.

4. Algorithm Implementation

Automatic code verification needs different e-test question structure which allows not just entering concrete answers but also encapsulating answers into questions, executing and obtaining some answer that can be checked.

The next figure (Fig. 3) presents alternative question structure, which is different from a standard quiz question.

Proposed question structure allows entering full program code and hiding some parts that must be rewritten by a student or serves like extra cases for avoiding hard-coding. Keywords that are allowed or forbidden to be used in student answers need to be considered as well.

This section presents an example that illustrates the approach and mentions a possibility to solve the problem by showing how code verification can be carried out for the code that can return the answer. A test question in Fig. 4 is a fragment with JAVA program source.

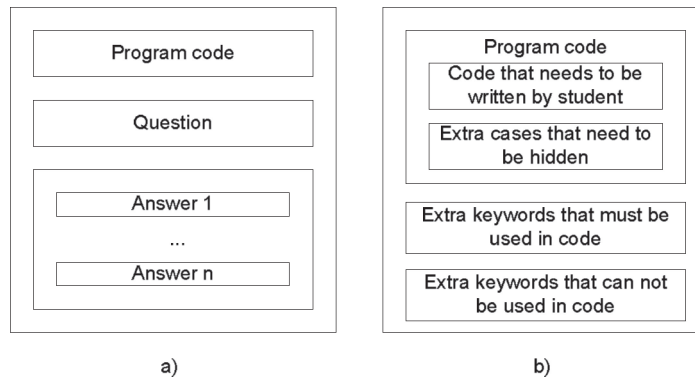


Fig. 3. Question structure meta-model: a) standard question; b) proposed question.

```

class A {
    /* Method orders String characters in alphabet order */
    String orderStr (String s) {
        /* To Do: Program code goes here */
    }
}
class B {
    public static void main (String a[]) {
        A obj = new A();
        if (!"acd".equals(obj.orderStr(new String ("cda")))) {
            System.out.println("False");
            return;
        }
        /* Hidden cases... */
        System.out.println("True");
    }
}

```

Fig. 4. Question example.

In this case a student gets a fragment of the program with a code which is already written. He/she has to read the code, understand it and extend the program with a predefined function. He/she also needs to write a class method for entering letters in a text area in the right order. Later a student sees some test cases that are intended for written function testing. The function has to correspond to the test cases. There are some hidden cases that a student does not see as the test cases serve as a protection against hard-coding. In this particular case a loop sentence FOR and a condition sentence IF can be used, and SORT function is forbidden.

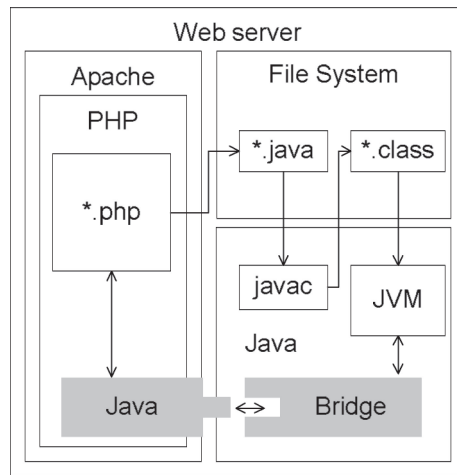


Fig. 5. Program code execution on the server.

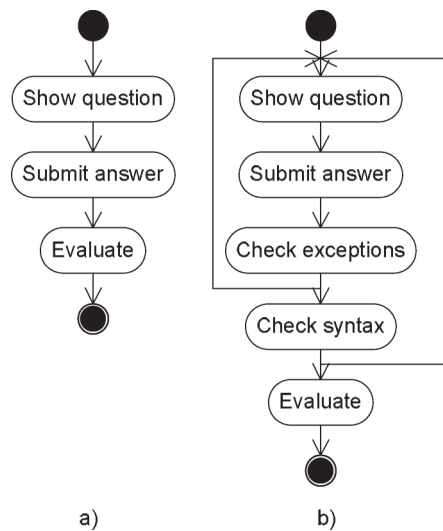


Fig. 6. Question checking a) standard question; b) proposed question.

The code execution on the server will involve saving student answer, which is his program, compiling the code with JAVA compiler (javac) and using Java Virtual Machine (JVM) from one side and Java-Bridge extension from PHP (see Fig. 5).

Checking and submitting answers is completed using two steps (Fig. 6). Carrying out the first step, the code is checked for exceptions (forbidden/allowable keywords, using system and file functions) and syntax errors. Checking syntax knowledge refers to the ability to fix errors within the written program code. There is no need to test student's syntax knowledge because all IDE provides code syntax checking. After that, the student can fix his/her errors and submit the functioning code for the evaluation.

Table 1
Decision table for answer evaluation

Conditions														
Program compiles (3 evaluation points)	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N
Forbidden keywords are used	Y	Y	Y	Y	N	N	N	N	Y	Y	Y	N	N	N
Only right keywords are used (3 evaluation points)	Y	Y	N	N	Y	Y	N	N	Y	Y	N	N	Y	Y
Result is correct (4 evaluation points)	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N
Actions														
Allow program bug fixing										X	X	X	X	X
Tell to remove forbidden keywords					X	X	X	X				X	X	X
Evaluation is possible	X	X	X	X					X	X	X	X		
Evaluation														
Point 1					+	+	+	+						
Point 2					+	+	+	+						
Point 3					+	+	+	+						
Point 4					+	+						+	+	
Point 5					+	+						+	+	
Point 6					+	+						+	+	
Point 7					+		+							
Point 8					+		+							
Point 9					+		+							
Point 10					+		+							
Result														
Sum	0	0	0	0	10	6	7	3	3	3	0	0	0	0

To eliminate question's evaluation problem when just two possible results are available, the decision table (Table 1) is proposed to make set of results larger. In this case four evaluation conditions are offered and all possible cases are listed. Decision table allows planning actions that may be done according those conditions and also evaluate every condition that is satisfied by the student. Three conditions are evaluated with points, only if forbidden keywords are used evaluation must be stopped because of the possible hacking. Obtained result range is wider with decision table.

5. Security

Server scripting languages like PHP, ASP or JSP are possible to be used for the implementation of tests. Such languages possess build in functions that allow executing C, JAVA or their own program codes and get an answer. There are many C or JAVA compilers available in the Internet and those languages are substantial seeking to check student knowledge on algorithm creation.

In all cases security remains the open question because a written program code may be harmful to e-testing server, that is why secure program execution becomes of key importance. Most work can be carried out during the phase of exception checking, because the checking of system usage and file functions must be limited. The next phase involves code saving and execution in a web safe directory, which is not reachable for web users.

6. Conclusions

E-testing has numerous advantages over traditional paper-and-pencil testing; however, it brings some challenges as well. The development of e-testing tools has become an important issue in the development of e-learning systems. This article analyzes the method for the creation of e-testing tools. The suggested method allows creating and completing tests for computer science students. It is also suitable for checking and evaluating test questions with freely entered program code, automatic checking of syntax correctness and the usage of right keywords.

This type of testing is a helpful tool for teachers as it reduces their work and provides students with evaluation of their answers. Moreover, e-testing would be a useful tool for employers, selecting and hiring programmers because this method allows testing their computer science skills, abilities and practical knowledge that is of key importance in this specialty.

References

- Bergstrom, B. (1992). Ability measure equivalents of computer adaptive and pencil and paper tests: A research synthesis. Paper presented at the *Annual Meeting of the American Educational Research Association*, San Francisco.
- Brusilovsky, P., Miller, P. (1999). Web-based testing for distance education. In: *Proceedings of WebNet*, AACE, 135–168.
- Brusilovsky, P., Nijhavan, H. (2002). A framework for adaptive e-learning based on distributed re-usable learning activities. In: *Proceedings of World Conference on E-Learning*, E-Learn. URL: <http://www2.sis.pitt.edu/~peterb/papers/ELearn02.pdf>.
- Georgiadou, E., Triantafillou, E., Economides, A.A. (2006) Evaluation parameters for computer adaptive testing. *British Journal of Educational Technology*, 37(2), 261–278.
- Gibbons, R.D., Weiss, D.J., Kupfer, D.J., Frank, E., Fagiolini, A., Grochocinski, V.J., Bhaumik, D.K., Stover, A., Bock, R.D., Immekus, J.C. (2008). Using computerized adaptive testing to reduce the burden of mental health assessment. *Center for Health Statistics*, University of Illinois, Chicago, 59(4), 361–368.
- Gierl, M.J., Jiawen Zhou, J. (2008). Computer adaptive-attribute testing: A new approach to cognitive diagnostic assessment. *Zeitschrift für Psychologie / Journal of Psychology*, 216(1), 29–39.
- Hardré, P.L., Crowson, H.M., Xie, K., Ly, C. (2006). Testing differential effects of computer-based, web-based and paper-based administration of questionnaire research instruments. *British Journal of Educational Technology*, 38(1), 5–22.
- Harnes, J.C., Kromrey, J.D., Parshall, C.G. (2001). *Online Item Parameter Recalibration: Application of Missing Data Treatments to Overcome the Effects of Sparse Data Conditions in a Computerized Adaptive Version of the MCAT*.
- Meijer, R.R., Palmer M.L. (1999). Computerized adaptive testing: Overview and introduction. SageJournals Online, *Applied Psychological Measurement*, 23(3), 187–194.
- Rudner Lawrence, M. (1998). *An On-line, Interactive, Computer Adaptive Testing Tutorial*. 11/1998, URL: http://EdRes.org/scripts/cat_review on 2008.11.23

Tonidandel, S., Quiñones, Miguel A., Adams, Ann A. (2002). Computer-adaptive testing: The impact of test characteristics on perceived performance and test takers' reactions. *Journal of Applied Psychology*, 87(2), 320–332.

Wainer, H. (2000). *Computerized Adaptive Testing: A Primer*. Lawrence Erlbaum Associates, Hillsdale, NJ.

S. Drašutis holds the position of a lecturer in the Department of Multimedia Engineering, Faculty of Informatics, Kaunas University of Technology, Lithuania. His current research interests are in distance learning systems, multimedia engineering and web development. He is a researcher in several international projects and author or co-author of 8 research papers.

V. Motekaitytė is a PhD student and major coordinator at the Department of Multimedia Engineering, Faculty of Informatics, Kaunas University of Technology, Lithuania. Her research interests include rationalization methods and provision of e-learning syllabus and methodical material, also e-testing methods, learning technologies, management and structure of learning systems, interactive e-learning environments, application of intelligent techniques and systems, software engineering. She is a coordinator and researcher in national and international projects and author or co-author of 14 research papers.

A. Noreika holds the position of a lecturer in the Department of Multimedia Engineering, Faculty of Informatics, Kaunas University of Technology, Lithuania. His current research interests are in multimedia based web systems development, multimedia engineering and computer graphics. He is a researcher and developer of several projects and author or co-author of 6 research papers.

Automatizuoto programos kodo testavimo metodas

Sigitas DRAŠUTIS, Vida MOTEKAITYTĖ, Algirdas NOREIKA

Informacinės komunikacinės technologijos, interneto galimybės ir prieigos taškų didėjimas skatina visuomenė perkelti dalį gyvenimiškų veiklų į virtualią erdvę, todėl gana sparčiai kuriasi e. bibliotekos, e. kultūra, e. pramogos ir, žinoma e. mokymas, kuomet siekiama pagerinti mokymosi efektyvumą mokyklose, kolegijose, universitetuose. Šiuolaikinis e. mokymas negali būti suvokiamas be e. testavimo, tačiau daugeliu atvejų testavimo įrankiai yra dažniausiai pritaikomi ir naudojami tik standartinėms/teorinėms žinioms patikrinti, panaudojant klausimus, apklausas, sulyginimus, pasirinkimus vieną/daug iš kelių variantų ir panašiai. Šiame straipsnyje aptariama e. testavimui skirtų sistemų funkcinių galimybių stoka ir pasiūlomas metodas besimokančiųjų, studijuojančių informatiką, žinioms įvertinti. Pasiūlytu žinių patikrinimo metodu sudaromos galimybės įvesti laisvos formos atsakymą (programos fragmentą), patikrinti programinio kodo sintaksę bei įvertinti parašytos programos teisingumą.