The Automatic Evaluation Testing

Tasks with graphics character

Martin Němec, Radoslav Fasuga
Department of Computer Science
VŠB – Technical University
Ostrava, Czech Republic
martin.nemec@vsb.cz, radoslav.fasuga@vsb.cz

***Abstract*—With the further development of computer technology, the non variant test question with automatic evaluation has been gradually applied at schools and universities, which positively reduces the work volume of teachers and demonstrates a proper way to test students. The primary aim of this paper is to describe the methods and algorithms for determination of equivalency and similarity rate of tasks' graphic outputs in learning management system. Among the analyzed types, there are tasks the result of which is a graph of corner-edge type, logic circuit, geometric construction or graphic behavior of function of one real variable. We performed more than 2,000 student's final tests that confirm the applicability of our solutions.**

Keywords-logic diagram similarity; geometric constructions similarity; function behavior similarity; automatic testing; e-learning

I. INTRODUCTION

Nowadays, information technologies are used more frequently in the educational process. There are several reasons for the phenomenon. The rising number of the computers and the Internet access creates a very easy way of getting information from the Internet and consequent eventual comparison of these information with other students. Use of computers in all methods of education helps to improve understanding of the created teaching materials and also allow students to involve and use several senses into the training process.

A Knowledge Testing is part of the training process. Nowadays different learning management systems (LMS) enable us to create tests with online assessment [1]. These tests are mostly textual and once we need to include the graphic questions (charts, diagrams, electrical circuits, etc.) the tests can be used in a very limited form or unusable at all [2][3]. Another possibility is using the external programs but they usually have some compilation issues with LMS and do not have automatic scoring feature [4][5]. They can mostly be used as tools for practical experiments only, not for testing.

Our aim is to introduce the tool that includes modules for processing some of the graphic tasks. These modules extend the LMS to the possibility of testing the graphic tasks and allow analyzing them automatically.

II. STATE OF THE ART

Defining the main educational goals is very important part of the whole educational process. The aim of education is a characteristic change of the student which should be generally achieved during a certain period of time (once the student has passed the training) [6][7].

In the most of the cases, the Bloom's taxonomy of educational objectives is considered as a basis. The original Bloom's Taxonomy divides educational objectives into three domains (Cognitive, Affective and Psychomotor).

The new revised Bloom's taxonomy (Fig. 1) is aimed just for a cognitive domain as the complex domain. This domain is prefered by lectors. The revised Bloom's taxonomy divides the educational goals into six basic groups [8][9].

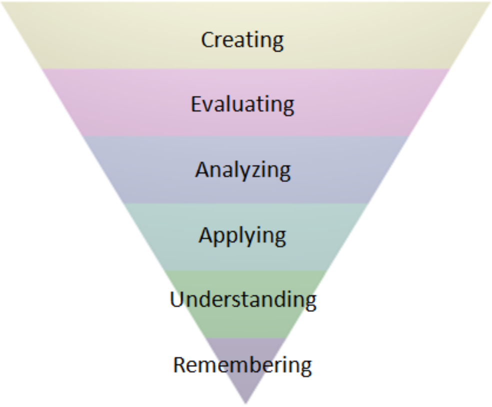


Figure 1. The new revised Bloom's Taxonomy

Bloom's Revised Taxonomy:

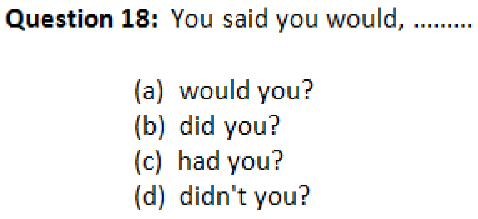
1. Remembering: can the student recall or remember information?
2. Understanding: can the student explain ideas or concepts?
3. Applying: can the student use information in a new way?
4. Analyzing: can the student distinguish between the different parts?
5. Evaluating: can the student justify a stand or decision?
6. Creating: can the student create new product or point of view?

ISBN: 978-1-61208-120-5

It is necessary to test and automatically evaluate more sophisticated knowledge than remembering only because we need to know whether the student can understand, apply, analyze, evaluate and create individual problems (Fig. 1).

A. *Testing*

Examination process is one of the most important parts of the whole educational system. Examination is used for checking the students' acquired knowledge, but it is also the direct part of educational process. Furthermore, the testing is taking advantage at all six points of Bloom's taxonomy.



Question 18: You said you would, ...

(a) would you?
(b) did you?
(c) had you?
(d) didn't you?

Figure 2. Sample of text variant question

Variant questions are the most common way of testing nowadays (Fig. 2). Mostly the way of testing is especially suitable for testing at the first area, such as memory testing of a small group of people. Memory testing in a big group of people by the method of variant questions is usually very hardly achievable, mostly impossible.

Another way of testing are simply made with questions that allow to the author to achieve the specific inquiry where students have to create their own answer such as number, vector, character string, etc.

The students have to make the solution by themselves not choosing it from the represented pictures as it happens in the case of the variant questions. Unfortunately, simply made questions are mainly suitable to present the result in the text form.

The special types of the pictures with graphical form are very infrequently or almost never used. The example of these kinds of questions may be assembled like a different schemes based on the principles of diagrams. Examples of such types of questions may be putting together the various schemes based on the principle of the charts. This includes the UML diagrams, ER diagrams, etc. An example of such a diagram is shown in Fig. 3.

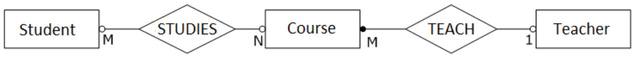
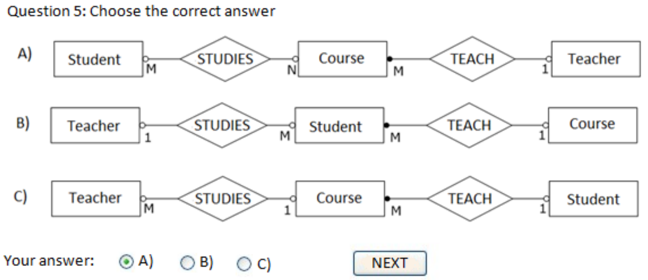


Figure 3. Example of E-R diagram

This type of graphic questions is quite like the variant answers. The student has to select the correct answer from several variants (Fig. 4). The variant tests are widely used today, but this type of testing is good for memory testing only, not for other Bloom's groups (Understanding, Applying, Analyzing, Evaluating and Creating) (Fig. 1). And these educational goals are more important than remembering.

When we want to test these educational goals, we need to make students creating their own graph, diagram, circuits, etc. And we want as well the results to be evaluated automatically.



Question 5: Choose the correct answer

Your answer: ● A) ○ B) ○ C)

Figure 4. Example of a graphic question with variant answers

Our aim was to prove that these types of graphics tasks can be automatically evaluated with a properly created algorithm and with support of software environment.

As each of the tasks may have different features and algorithms, the individual test modules have been created, which can be practically applied and tested in education process.

III. TESTING MODULES

Currently, we have to verify the functionality of the developed and tested modules, which can be divided into four categories. The result of this work is to verify the use of the test modules in practical teaching process.

Students have to create their answer using the module with graphics interface in a Learning Management Systems (LMS), like Blackboard [10], WebCT [11], or Moodle [12].

A. *Tasks with graph result*

In practice we face the tasks of which the result solution is a graph (node-edge type). This means that the result is a graphical task consisting of a variable number of nodes and edges of different types.

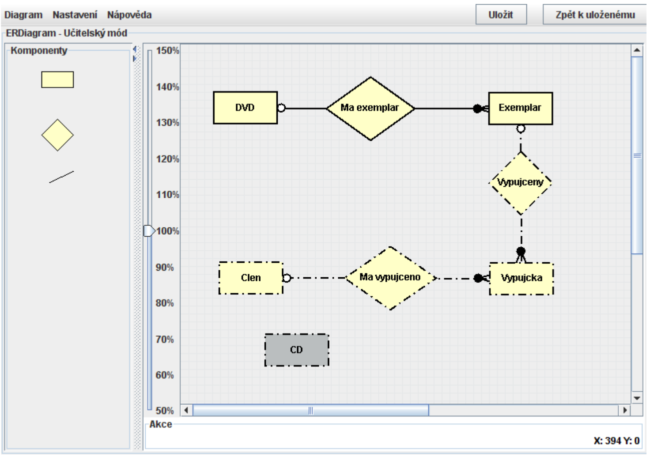


Figure 5. Screen shot of testing module for ER diagrams

The example of the task can be database's theory in ER-Diagram, UML diagram, the diagram in SW engineering, etc. The problem arises when we need to check these two solutions and figure out if the solutions are equal or similar to each other. The basic problem of comparing these tasks is their independency on position in a graphical placement of chart elements.

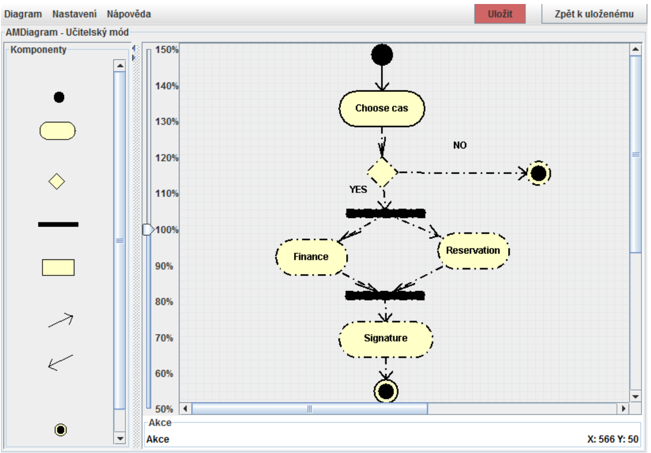


Figure 6. Screen shot of testing module for UML diagrams

Two different-looking graphics tasks can be logically identical. One of the goals of this work is to find out methods and their corresponding algorithms, which allow evaluating these type-task results and comparing the resulting degree of similarity between two tasks results.

The main principle of automatic evaluation is to separate a result, which is represented as a node-edge diagram into two sets. The first set includes nodes and the second set includes edges.

Some limits enabling identification of particular objects are essential to be able to evaluate the tests results automatically that´s why the author has to create a set of names that contains a variety of node names. Not only the set may include the correct nodes, but also the names that are not need for a correct result; these names can be used for creating more difficult questions.

As the first step, the algorithm evaluates set of nodes with their attributes between original graph A (created by a teacher) and evaluation graph B (created by a student). After the algorithm evaluates the nodes, it takes the set of edges. All edges have self attributes and two nodes. Once we have all the sets evaluated, we can calculate the final similarity.

We use the following equation for evaluating (1), where n(x) is the set of edge weight, e(x) is the nodes weight, O(X) is the number of elements and d(A,B) is the measures of graph similarity [10].

$$d(A,B) = 1 - \frac{\sum_{i=0}^{p}\sum_{j=0}^{r}\left|(n(u_i) - n(v_j))\right| + \sum_{i=0}^{q}\sum_{j=0}^{s}\left|(e(u_i) - e(v_j))\right|}{\max(O(A), O(B))} \qquad (1)$$

The sample modules for testing graph results are shown in Fig. 5 and Fig. 6.

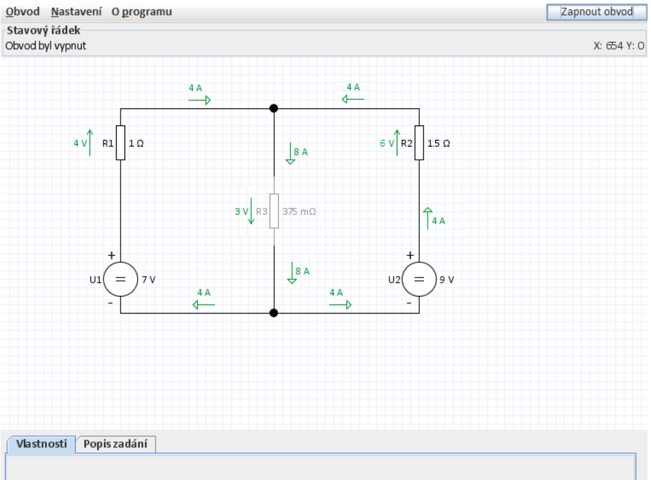
B. *Tasks for testing circuit simulation*

In some areas, such as logic circuits, electrical circuits, etc., a similar problem occurs as with the graph. Two circuit diagrams are given and our aim is to compare these schemes to find the accurate connection.

In this case it is not possible to use a similar principle as the comparison in the previous case, as two circuits having the same characteristics, but different connection may exist.

According to the selected type of scheme, there are various circuit elements: resistor, coil capacity, power, etc. are for electric circuits. The logic gates have different types (none, AND, OR, NAND, etc.).

However, accuracy of the connection depends on accuracy of the circuit input and output.



Obvod byl vypnut

Figure 7. Screen shot of testing module for electric circuits

Nowadays, there are professional tools for simulating the circuits including the setting characteristics of separate variables on the relevant parts exist, unfortunately, the tools are not intended for learning purpose and require specific knowledge of the field. As a result they can't be used for familiarizing the students with the subject.

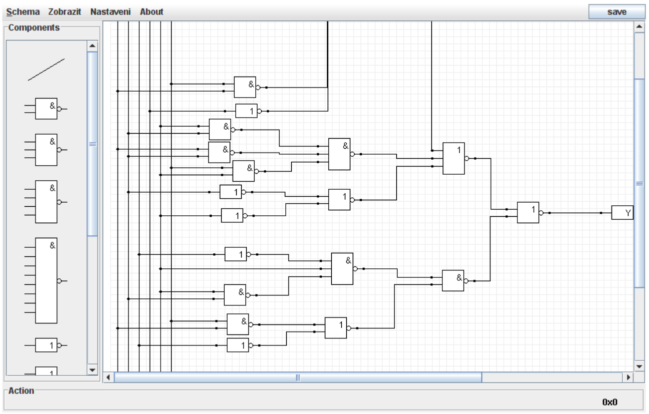


Figure 8. Screen shot of testing module for logical circuits

The main principle of evaluation is a conversion of the graph into the truth function for logical circuits [14] or function for analyzing electric circuits [15]. We evaluate and compare the student's final function and teacher's function.

We use the equation (2) for evaluation, where y_i is the value of the student's function (truth table) and z_i is the value of the teacher's function. The aim is to meet the target primarily on opportunity of automatic testing and valuation of the received information.

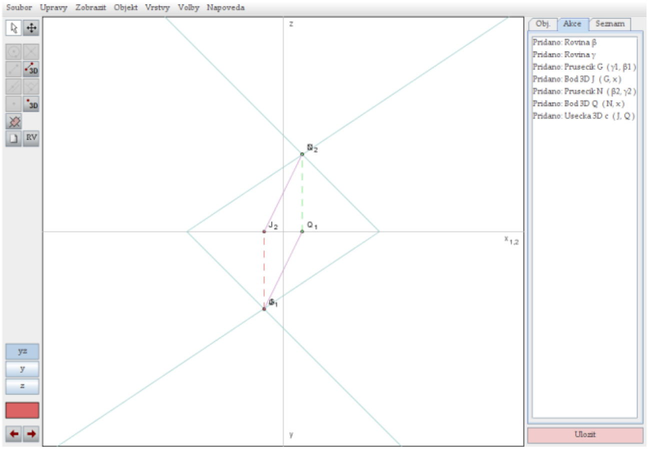
$$d(A,B) = 1 - \frac{\sum_{i=0}^{2^k-1} \left|(y_i - z_i)\right|}{k} \text{, for } k{=}max(n{,}m) \qquad (2)$$

Our module can not only test electric circuits and combinational logic circuits, but also sequential logic circuit [14]. The sample modules for testing circuit simulation are shown in Fig. 7 and Fig. 8.

C. Tasks for geometric construction

For teaching subjects such as mathematics, geometry, descriptive geometry, computer graphics, etc., it is necessary to compare the tasks based on the geometrical structures, for which several ways of solution exist. It means that our next researched area is the comparison of equivalence of geometric structures.

Design task represents a set of graphical objects (point, line, plane, etc.) and a number of construction methods (e.g. calculating the intersection of two lines). The whole design task can be divided into following sages – the given objects, the results of single design steps, and the result – the final objects.



Pridano: Rovina β
Pridano: Rovina γ
Pridano: Prusecik G (γ1, β1)
Pridano: Bod 3D J (G, x)
Pridano: Prusecik N (β2, γ2)
Pridano: Bod 3D Q (N, x)
Pridano: Usecka 3D c (J, Q)

Figure 9. Screen shot of testing module for descriptive geometry

Examples of such types of tasks are geometrical constructions in descriptive geometry and computer graphics, for example the constructions in Monge projection, in axonometry, in oblique projection, etc.

We can evaluate similarity with equation (3), where $v_{A,i}$ is the set of student's construction steps and $v_{B,i}$ is the set of teacher's constructions steps. In our case we focused on the Monge projection only because general description of each type of projection would be very extensive and it is not necessary for the resulting functionality [16].

$$d(A,B) = 1 - \frac{\sum_{i=0}^{n} \left|(v_{A,i} - v_{B,i})\right|}{n} \qquad (3)$$

The sample module for testing geometric procedures is shown in Fig. 9. We prepare next modules for other projections (axonometric projection, perspective projection and others).

D. Graphic behavior of function of one real variable

Investigation of graph of the function defined by the functional quote y = f (x) is one of the mathematics subjects. In this case the task is to calculate the function properties based on the function quote (landmarks, the intervals of monotonicity, etc.) and use these properties to outline the resulting graph of this function [17].

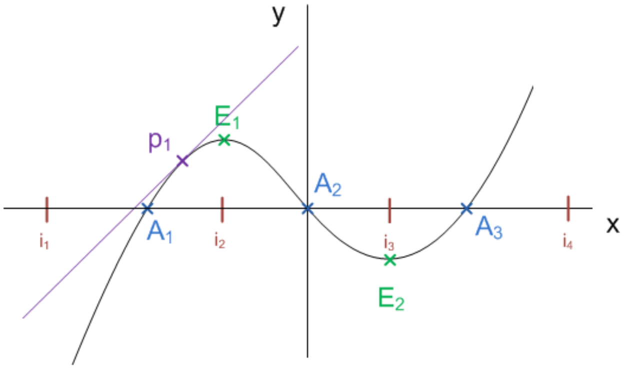


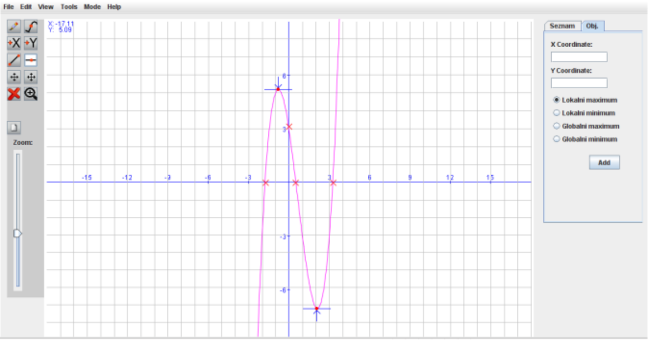
Figure 10. A Graph of example function

The Fig. 10 shows an example of the real function of the function f with some elements (points A1, E1, p1 tangent, the end points of major intervals i1, etc.) demonstrating the basic properties of the investigated function. In general, there are the following characteristics such as roots, extreme, inflection points, boundedness etc. The attributes are used to describe the three basic types of elements.

Basic elements for characterization:

- Point – roots, intersections, extremes, etc.
- Interval – interval of function increasing or decreasing, etc.
- Lines – asymptote, tangent line, etc.

The determining of the graph is based on the properties, calculated from the specified function y = f (x). Student will investigate the resulting graph and draw it manually, based on the properties.



X Coordinate:

Y Coordinate:

● Lokalni maximum
○ Lokalni minimum
○ Globalni maximum
○ Globalni minimum

Figure 11. Screen shot of testing module for graphics function

$$d(A,B) = 1 - \frac{\dfrac{\sum_{i=a}^{b} \left|(f(i) - (m_i \pm \delta))\right|}{|b-a|} + \dfrac{\sum_{i=0}^{n} \left|(p_i - (r_i \pm \sigma))\right|}{n}}{2} \qquad (4)$$

The evaluation equation compares distance of the accurate graph with the student's graph approximate curve (4).

The final level of similarity is calculated as the deviation of the student sketched curves, comparing actual progress on the specified interval <a,b> with accuracy δ. If the graph contains a set of characteristics (roots, intersections, extremes, etc.), we have to compare this set too. The sample test module for testing the function is shown in Fig. 11.

IV. Module Properties

All generated test modules can be run in three basic modes:

- Author's Mode - The module runs to allow the author to create specifications and sample solutions of the tasks.
- Student's Mode - This mode enables students to create their own final solution.
- View Mode - In this mode the module is configured to display the final solution without the possibility of intervention into the solution. Student and tutor can view the student's created solutions. It can also display the correct solution to the student (e.g. during self-testing process).

The author has several kinds of evaluations:

- Absolute evaluation – server evaluates full resulting and evaluated task as absolutely correct or not.
- Percentage evaluation – author can define percent evaluation and server evaluates proportional result.

Primary benefits include the following:

- The proposed algorithms can be tested by the students not only for memorizing the learning material but practically at all levels according to Bloom's taxonomy (remember, understand, apply, analyze, evaluate and create).
- Everything can be processed using LM systems electronically and transparently, which also allows you to get a feedback (which part of education is the most difficult for students).
- Tutor has no need to evaluate the tests results manually anymore. The modules and process will evaluate the tests automatically and display the correct results to the students.
- Students accessing from the Internet can verify their knowledge anywhere and any time. Moreover, as it has already been mentioned, testing is a direct part of the process of teaching.

A. *The use of testing modules in LM systems*

Test modules is planned to be implemented in the LM systems and design of the modules is adapted for this installation [10][11][12].

It is necessary to define the format of data for their transferring between different applications and server, processing and storing. This format should be adaptive for possible changes, suitable to transfer data via Internet and also for processing within the test modules.

Use of LM systems has the essential advantages:

- Easy creating and defining of the tests.
- Clarity on the test details (when, who performed the test, score, duration, number of attempts, etc.).
- Easy storage and archiving of all activities using the database.
- Statistical evaluation of past tests that can help to reject the inappropriate tests (a small percentage of correct solutions).

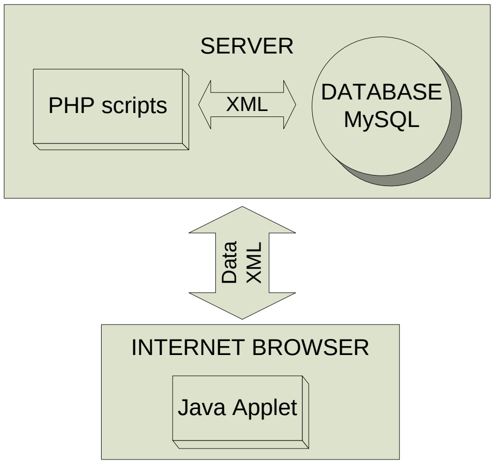


Figure 12. Modules communications

All communication and data processing scripts are created in PHP but they can always be modified for other languages, e.g., ASP. Setting of the modules properties using a few basic parameters is part of module building-in process into the LMS. With these parameters the basic features such as an identification key chart type, mode of diagrams, and way of storing and retrieving data are set. The build-in module with setting is shown in Fig. 12.

ISBN: 978-1-61208-120-5

Current test counts are written in Table I. We have two types of tests. Final test must be completed by a student before in the end of a semester. Practical test may be completed by a student throughout the whole semester several times.

TABLE I. TABLE OF TEST COUNTS

Number of tests counts		
Type of tasks	***Practice test count***	***Final test count***
Module for testing ER diagrams	1000	1000
Module for testing UML diagrams	1000	1000
Logic circuit	200	50
Geometric construction	100	20
Graphic behavior	100	30

In the testing, we performed more than 2,000 real final tests in education that confirms the applicability of the various modules in teaching.

V. CONCLUSION AND FUTURE WORKS

Our goal is to show that the opportunity of automatic testing is not limited on using variant and simply made questions. It is possible to carry out the automatic check of graphic tasks if the objects are properly described.

The design and implementation of the graphic testing modules for chosen areas demonstrates its big contribution to training automation sphere. This type of education is very practical a useful for both tutors and students. With a help of these modules, we can extend our possibilities of automatic evaluation. Now we don´t need to use the variant questions only because we can require such type of questions as “Create the diagram”, “Correct the electric circuit”, “Connect logical circuit”, “Draw construction of Monge projection” and others.

In the future we are planning to work over other types of graphic problems to demonstrate opportunities of such a special type of tasks.

REFERENCES

[1] H. T. Huo, "Study on the Function of Test Paper and Questionnaire in Moodle Course", Proc. International conference on modeling and simulation, May 2009, pp. 271-274.

[2] Z. Xiaochun, Z. Bo, L. Juefeng, and G. Qiu, “A test automation solution on GUI functional test”, IEEE Conference on Software Maintenance, July 2008, pp. 1413-1418.

[3] A. Memon, “An event-flow model of GUI-based applications for testing”, IEEE conference on Software Testing, Verification and Reliability, Vol. 17, September 2007, pp. 137-157.

[4] H. Uzunboylu, "Using dynamic geometry software Cabri 3D for teaching analytic geometry", Innovation and kreativity in education, vol. 2, 2010, pp. 1385-1389.

[5] H. Pesman and A. Eryilmaz, "Development of a Three-Tier Test to Assess Misconceptions About Simple Electric Circuits", Journal of educational research, vol. 103, 2010, pp. 208-222.

[6] B. Bloom, "Taxonomy of Educational Objectives, The Classification of Educational Goals. Handbook I: Cognitive Domain.", New York: David McKay, 1956.

[7] A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom’s Taxonomy of Educational Objectives, edited by L. W. Anderson and D. R. Kraftwohl _Longman, New York, 2001.

[8] C. Johnson and U. Fuller, "Is Bloom’s Taxonomy Appropriate for Computer Science?", Proceedings of the 6th Baltic Sea Conference on Computing Education Research (AICPS ’06), vol. 276, 2006, pp. 120-123.

[9] T. Ho and T. Lee, "Teaching Thinking Skills In E-Learning - Application Of The BLOOM’S TAXONOMY", ITE Teachers' Conference, 2004.

[10] Blackboard Inc, Blackboard Course Management System [online], [cit. 2010-12-10]. Available: http://www.blackboard.com/.

[11] WebCT, Course Management System [online], [cit. 2010-12-10]. Available: http://www.webct.com/.

[12] Moodle, Course Management System [online], [cit. 2010-12-10]. Available: http://moodle.org/.

[13] J. Fournier, "Graph Theory and Applications", France, 2009, ISBN 978-1848210707.

[14] J. Cavanagh, "Sequential logic: analysis and synthesis", CRC/Taylor & Francis, 2007, ISBN 978-0849375644.

[15] K. Charles and S. Matthew, "Fundamentals of Electric Circuits", 4th Edition, McGraw-Hill, 2009, ISBN 978-0077263195.

[16] E. Watts, "Descriptive Geometry", Watts Press, 2008, ISBN 978-1443730006.

[17] L. Graves. "The Theory Of Functions Of Real Variables", Second Edition, Dover, 2009, ISBN 978-0486474342.