

TestTool: Web-based Testing, Assessment, Learning

Kazys BANIULIS

*Computer Networking Department, Kaunas University of Technology
Studenty 50–416, LT-3031 Kaunas, Lithuania
e-mail: kazysba@pit.ktu.lt*

Vytautas REKLAITIS

*Graduate School of Information Systems
University of Electro-Communications, Tokyo, Japan
e-mail: vyta@is.uec.ac.jp*

Received: May 2002

Abstract. In this paper we look at the testing and assessment from the perspective of the learning outcomes we are going to assess. As the TestTool was designed and implemented at the technological higher education establishment it was meant as a web-based medium to test and assess future engineers' capabilities to solve rather simple constructive tasks. For such a purpose typical testing questions with predefined answer alternatives appeared to be not effective. We introduced a 'graphical construct' question type, which accommodates constructivist learner-centered style of learning. We discuss how this type of question extends TestTool functionality by providing instructional designs of questions from "Data Structures" module, which has been delivered for Computer Sciences undergraduates. TestTool implementation along with considerations regarding its wider use is briefly discussed in the final part of the paper.

Key words: e-learning, web-based assessment, testing.

1. Introduction

There is a range of web-based services to support learning, but in educational settings most commonly used are access to structured learning resources and Internet communication facilities. Our approach to network-based education focuses on effective use of standard or proprietary software tools to enhance and improve the web-based individualized self-instructional mode of learning, grounded on the constructivist model of education. What we are seeking is a means of computerized assessment appropriate for learning through knowledge building. In other words, we look for e-learning facilities, which allow learners to manipulate with subject matter elements in the process of problem solving, and also enable tutors to trace students' actions and to assess their capabilities according to the task solved. The TestTool package (Baniulis *et al.*, 2000) designed for self-learning and assessment is an attempt to approach this problem.

Testing is being increasingly used in higher education institutions. The survey (Stephens, 1997) made at Loughborough University names the reasons for wider scale use of Computer-Assisted Assessment (CAA) in higher education institutions of UK and provides most typical cases of testing for assessment. CAA is seen as particularly appropriate for knowledge-based subjects with recall and recognition skills predominating. Meanwhile, learning of any technical or technological discipline is based mainly on knowledge building via real understanding how process or construction works as a whole (Squires, 1999). The real understanding comes up through achievement of certain level of subject knowledge, as well as through developing skills applying that knowledge when solving tasks. CAA environments currently in use (Crofts, 2000; Stephens, 1997; Vantaggiato, 1998) help mainly to assess level of knowledge, but have limited skills development and assessment possibilities. The TestTool gives an opportunity for creative instructors to explore and experiment with instructional designs aimed at assessment of skills and understanding.

This paper is structured as follows. In the next section we discuss on most critical conceptual issue for computerized assessment, it is the testing questions types. We look at this problem from the perspective of the learning outcomes we are going to assess by testing. Then we provide functional description of the TestTool in terms of UML use cases. Further in this paper we will discuss only the test authoring use case as most relevant to learning by introducing so called 'graphical construct' question type. This type of question allows instructional designs, which extend the TestTool functionality beyond a typical testing for certain level of knowledge. We discuss this issue by providing instructional designs of questions from "Data Structures" module, which has been delivered for Computer Science undergraduates by making use of TestTool extensively. TestTool implementation along with considerations regarding its wider use are briefly discussed in the final part of the paper.

2. A Point of View on Testing Question Types

The computerized assessment unit usually is the combination of a question along with prescribed form to input an answer, and therefore an answer evaluation method implicitly bound to that question. The survey (Stephens, 1997) shows that the range of question types being used in CAA settings is rather narrow with a great predominance of multiple choice questions (MCQs). Other types of questions, such as matching, hot spot-single marker, matrix list, multiple response, pull down list, are being increasingly used in advanced CAA systems such as Question Mark, TRIADS (Boyle *et al.*, 2000; MacKenzie, 2000; Questionmark perception product information, 2002; TRIADS, 2002).

In this paper we look at question type taking into consideration not only the delivery form and style, we are also focusing on the answer evaluation method. From this perspective, all MCQ-alike types, which require the student to select or input a correct answer from a set of predefined alternatives, may be brought to the same category. Such questions, though realized through different interfaces, deploy the same answer evaluation mechanism and therefore have a similar formative influence on learning outcomes as MCQs.

But, are the varieties of multiple choice question forms enough to test and assess for profound knowledge, skills and proficiency? The question like this is greatly relevant to higher education environment, where distant learning courses gain more and more popularity. Our answer to the question fully coincides with the title of the paper "Computer Aided Assessment must be more than multiple-choice tests for it to be academically credible" (Davies, 2001). The only way to achieve academically credibility of testing lies in radical changes and introduction of new forms and types of test questions. Obviously, that leads to the construction of a new CAA engines capable to analyze and evaluate/estimate the more complex forms of answers provided by students. Such systems certainly are more complex in terms of implementation, authoring of test questions, usage, maintenance and so forth. But, along with advancements in ICT there is no other way than to distinguish between the use of technology for knowledge construction versus knowledge memorization and reproduction.

Looking at types and forms of the test questions, it is important to be clear what learning outcomes we are trying to assess by selecting one or another type of question. The prominent and widely recognized B. Bloom's taxonomy (Anderson *et al.*, 2001) provides a useful structure in which to categorize test question types as regards learning outcomes. The taxonomy gives the six cognitive categories – recall, comprehension, application, analysis, synthesis and evaluation. When looking at learning in the context of technological means the number of categories is sometimes being reduced, but in any case, however, it is commonly admitted that MCQs covers only the first category. MCQs are relevant to assess how learner acquired the knowledge elements and/or concepts being used within a subject domain. By using this category of questions, however, it is impossible to assess learners' skills and performance in task solving situations, when knowledge elements are to be meaningfully applied to build more complex element of the domain being taught.

Authors of the instructional design standards for quality on-line courses (Standards, 2002) classify instructional design units according to the Performance/Knowledge (P/K) type. The instruction unit that has a particular learning objective must be of appropriate P/K type. Applying this standpoint to assessment question type we find out that majority of question types being used in current CAA systems are targeted to assess most simple P/K types, classified as recall facts and elements, recall and identify concepts. We argue that more complex P/K learning units such as capability to apply concepts or perform tasks should have a corresponding assessment question types with more sophisticated answer analysis mechanisms. It should be anticipated, however, that we do not oppose to the use of multiple choice questioning as a method to solidify one's knowledge regarding subject matter elements and main concepts.

The TestTool assessment engine along with traditional types of questions includes a new question type addressed to assess a real problem understanding rather than recalling and looking for correct answer among a few predefined alternatives. We call this type of question as 'graphical construct' (Baniulis *et al.*, 2001). The concept of the 'graphical construct' question is in giving a student the task to create or assemble a correct construction from a set of objects spread in constrained area on the screen, called a 'graphical panel'. The content of a test question in this case is to build up a correct construction

from the objects given, avoiding a selection of the correct answer from the predefined ones. Behind the scene there must be a fully correct construct created by the question author, which is used by assessment engine to evaluate the answer submitted by student.

Actually the idea itself to do so is not new, but its implementation in practice is much more difficult if to compare with implementation of MCQs using HTML, CGI, Java scripts or any other web application programming interface (API). The 'drag&drop' type of question is most similar to the graphical construct. The 'drag&drop' semantics and suitability for creation of more comprehensive instructional designs for assessment greatly depends on the software being used to implement a method for answer evaluation. If only HTML and simple scripting languages are used, then only trivial forms of 'drag&drop' can be designed. For instance, dragging labels into marked positions on the static HTML form differs a little from traditional forms of MCQs if to look from the perspective of performance/knowledge type we are going to attain. At the other end of the range of software tools there are Macromedia Flash Technology (Macromedia, 2002; Standards, 2002) which provides a powerful tools for creation of various 'drag&drop' templates. In general, a software engineer having a task to implement the concept of the graphical construct question actually must trade-off between meaningfulness of the test question and possibility to check automatically the correctness of the answer to it. We came up to the compromise in our implementation of the graphical construct for TestTool making it as interactive Java program.

3. TestTool Use Cases

In this paragraph we look at the use cases for three categories of users: author, administrator, and learner. As it is usually in modeling, those three user categories should be much more anticipated as a roles rather than individuals.

An author uses TestTool authoring subsystem to create a question variant according to the particular instructional design with some assessment objective in mind. He either starts from creation of completely new variant, or downloads existing variant for editing. Every new question variant should be uploaded into the common variant's repository assigning an original name, in accordance with a naming rule chosen for given subject topic, chapter or course.

Having created a certain amount of variants an author may create a question by assigning one or more variants to that question. Only one of them will be randomly selected and presented to the learner when running the final assessment test. The use case 'Create question' assumes linking variants into the one the question structure, whereas 'Destroy question' stands for takedown that structure retaining variants untouched. There is the separate author's activity 'Delete variant' (Fig. 1).

Author's role in this use case is in great extent identified as role of instruction designer and therefore is focused on dealing with questions. The grouping of questions into the test is an activity assigned to the administrator. Administrator's role is split into three packages (Fig. 2).

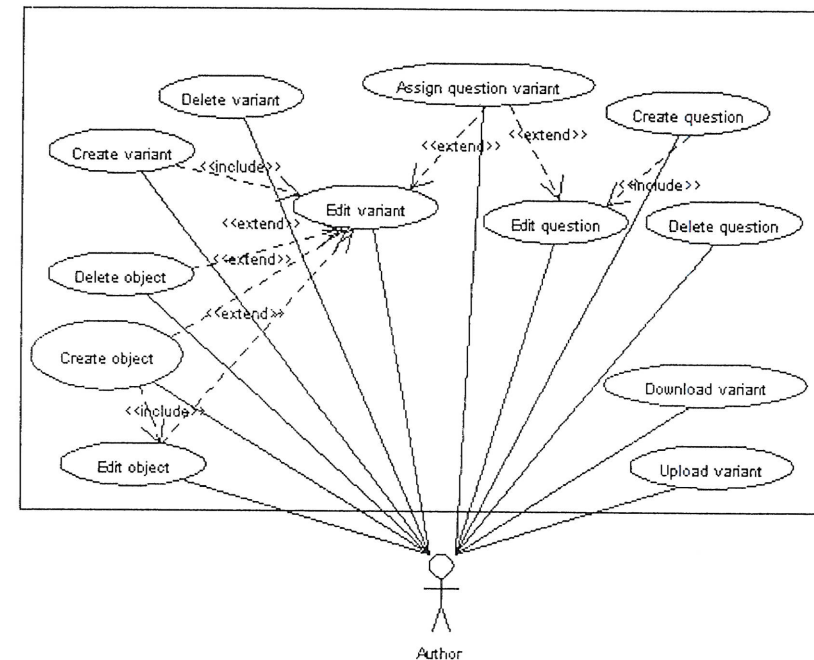


Fig. 1. Author use case diagram.

User management is thought as registration of new TestTool users. Every user receives authentication parameters and role. For a new user a learner's role is assigned by default. For the new author or new administrator the initial role must be changed respectively, whereas a learner must be assigned to one or several learners' group.

The test and exam management use case package models a creation of tests and exams. Test is viewed as linked list, which consists of arbitrarily chosen questions from the repository. Exam is a special kind of test, which receives additional restrictions on access and can be assigned to only one group of learners.

Typical administrator's role in course management is to create a course and to assign tests and student groups to that course. Thus a course may be understood as a set of several tests, which belongs to one subject module. Obviously, one particular question may be included into different tests.

Course management use case also reflects administrator's capability to view accumulated testing results and statistics. It is particularly important for instructors to have ability to view and discuss exam results together with individual learners. On the other hand, more general statistics about group performance when answering one or another test question allow instructors to estimate the instructional design of that question.

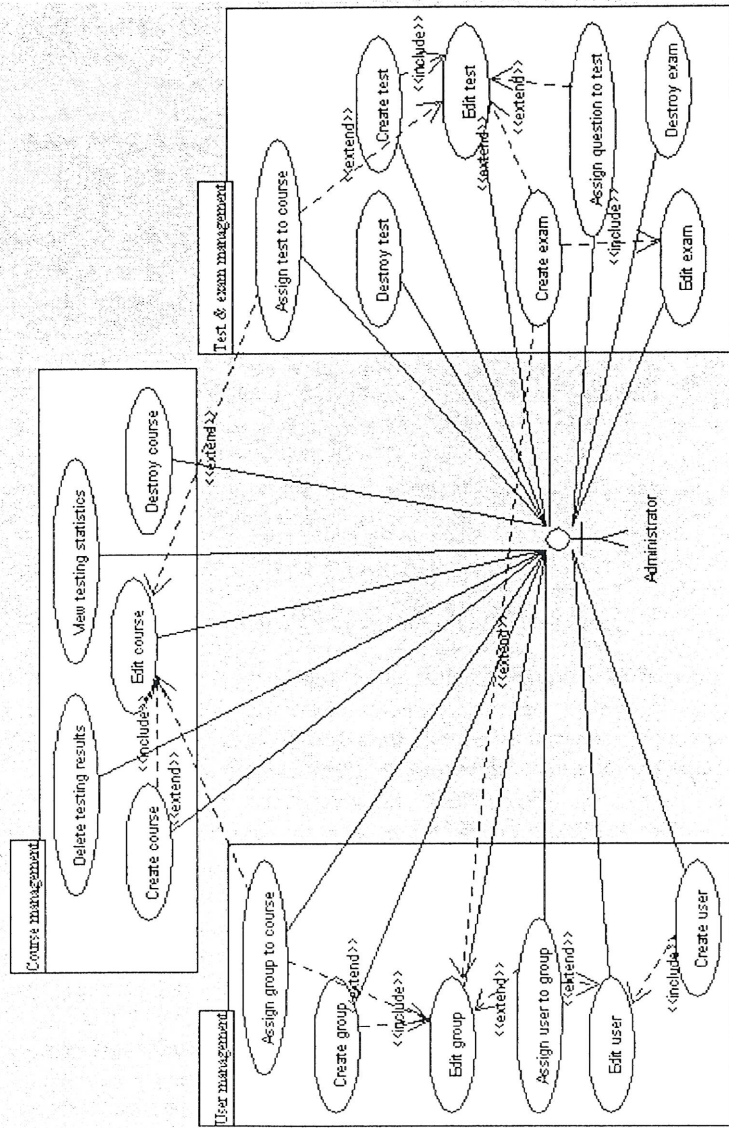


Fig. 2. Administrator use case diagram.

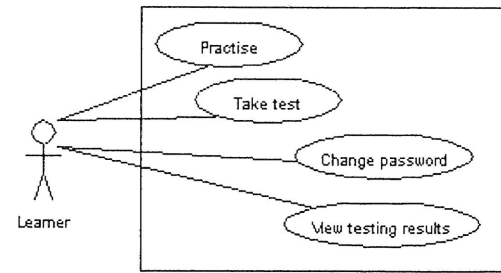


Fig. 3. Learner use case diagram.

Learners use case diagram shows that for learner two TestTool use modes are available – practice and testing. Practice actually is learning when student is able to take a tests assigned as many times as he/she likes and to view the assessment scores gained for each question. In this mode a learner is able to look through all equivalent question variants if such are included into practice test. Testing use case represents taking an exam test. Naturally, learners may access exam only once within the given time interval between two dates and they should do that in the classroom. Every individual session is being traced and student performance parameters are being recorded. For instance, if student disagrees with final evaluation grade calculated by assessment engine, all his answers can be retrieved from database and reviewed by tutor.

Logically ‘Test and exam management’ activities are continuation of test authoring and represent the administrative role in making of single test or exam. Both test and course management use cases complement authoring in completion of tests and subserve each other in an assessment environment creation. The roles of author and administrator overlap and often are being performed by one person, either lecturer or course tutor.

In the next paragraph we will focus mainly on instructional design and authoring of the graphical construct question variants because making use of this type of question may help to assess learners’ capabilities to solve tasks by applying knowledge being taught.

4. Test Question Authoring

The question authoring subsystem of the TestTool is a facility to create, edit and store a question variant. It functions as WYSIWYG (What You See Is What You Get) graphical editor and it is implemented in Java. When editing, editor’s window is split into two parts. Left-hand side is the area where question elements are to be put and task situation created. Any object put on this side also appears on the right-hand side area and may be edited there as it should be in the correct answer. For instance, if author is going to create a multiple choice question variant using list box form, he must to mark a correct answer only on right side.

In general, six types of text objects (text field, text box, list box, combo box, radio group, label) may be used for creation of various forms of questions with predefined

answer alternatives. The label is the simplest form to display question as a text, but also audio and video files can be used for question presentation. RealPlayer browser plugin that supports streaming technology is used to play video or audio files.

In order to create graphical construct variant a graphical panel option should be chosen from the 'New graphical object' menu. A panel serves as a container for all other graphical objects, which may be used when creating the graphical construct. There are five graphical objects – line, text string, rectangular, oval and image file. By editing properties of the object various attributes, like color, font type and size for texts, may be set for that object. After that they may be grouped into one item.

The instructional power, as well as a weakness, however, of the graphical construct resides in three other attributes. They are 'movable/non-movable', 'level of deviation' (applicable only for movable objects) and 'interchangeable/non-interchangeable'. Non-movable objects are put in their fixed positions by the task author, while movable objects may be dragged into any place within the panel. Only movable objects can be assigned 'level of deviation' attribute, which defines allowable deviation from the fixed position. If a few graphical entities of the same type may be fit into one construction place, they must be marked as interchangeable, i.e., they may take each other's position.

Thus, the power of a graphical construct is hidden in the necessity for learners to perform meaningful constructive actions using graphical items, which were laid out on graphical panel by tutor and comprise an instructional task situation. As an example, we describe here the content, instructional design and authoring of the task from "Data Structures" course.

The task is to insert a new element into AVL type tree retaining a given critical length of subtrees. This task may be included into test being used for practice and acquirement of skills, and equally may be used for assessment. The initial AVL tree is presented on Fig. 4.

A student should gain a working knowledge regarding transformations of a tree caused by insertion of a new element. For a given tree with a predefined critical lengths of subtrees a transformations to be done depend on the value of the node to be inserted. Two resulting trees received after insertion of 1 and 5 respectively are shown on the Fig. 5. This circumstance enables instructor to create easily a number of different task situations, whereas a learner is motivated to gain skills of dealing with a given kind of tree through

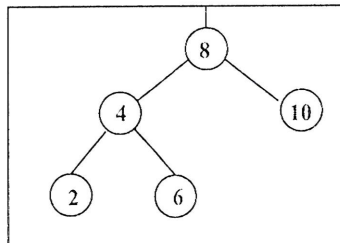


Fig. 4. The initial AVL-type tree.

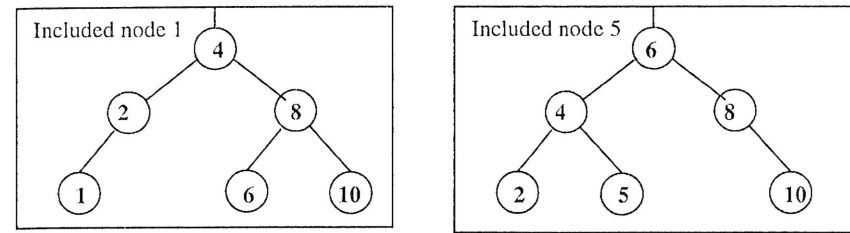


Fig. 5. Two resulting AVL trees.

the situation analysis including the search for relevant intermediate graph representations, which lead to the solution.

The instructional design of this task as a graphical construct question is shown on Fig. 6. This is authoring screen view, when author has simultaneously two panels displayed and is able to edit both of them. Left side is for creation of the task situation and right part for the correct solution. Note, the task situation contains drawing of the initial tree, a set of elements, which may be useful to build an answer, and circle simply just gives a sign to where not used elements have to be moved before submission of the correct answer. A tutor on the right-hand side of the panel must also create the correct solution. Every graphical object attains 'moveable' attribute automatically if its position is different on task and answer panels.

In self-learning or assessment mode only a task situation is displayed for learner and he is asked to reconstruct a given situation into answer by changing positions of objects. After submission of the answer, its correctness is being defined by checking absolute locations of the objects in the graphical panel. As it could be already understood, such a method is the main weakness of the graphical construct. It is because a student may drag any movable object into approximately correct location, but it could be 'not enough a bit' to hit a fixed position for this object defined by author taking even into consideration a

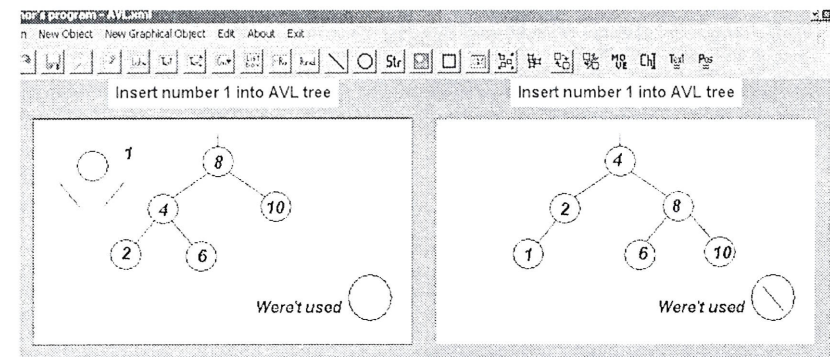


Fig. 6. AVL tree task instructional design.

value of deviation level. How to avoid ambiguity in automatic graphical answer interpretation is an issue of great importance. First of all, learners must be instructed to perform graphical construct tasks very carefully. In our practice that was achieved very quickly and easily, because TestTool has been used not only for final testing and examination, but also for learning and self-practice. For instance, a number of tasks related to a certain type of data structures have been created for "Data Structures" module and offered for students as practice tests. Taking them students not only acquire basic knowledge and skills, but also learn the finer practical points in dealing with graphical construct questions.

On the other hand, the risk of ambiguity may be considerably reduced by careful instructional design. An illustrative example is given on Fig. 7, where the task of insertion of a new element into B-tree along with correct solution is depicted. Like in the previous AVL tree case, the insertion causes re-building of the whole tree and answer construction process consists of rather complex sequence of actions. In order to reduce ambiguity in answer construction in this case graph node object is supplied with concatenation points. It does not make any influence on task content, but help learners to build consistent answer avoiding objects placement errors.

From the perspective of learning the two examples from DS course demonstrate that our intention is to assist students in understanding of and in acquiring skills in dealing with various data structures. Authoring of graphical construct tasks is quite simple process if tested and validated instructional designs are being used for creation of question variants. From pedagogical and methodical points of view the examples give also a clue what instructional designs may be created for one or another topic. In practice, the authoring is iterative and incremental process, during which a visual task situation should be created carefully choosing components and their attributes. As sooner the task is taken for practice the better. Learners help to find weak and ambiguous points in a design of the task. After a few iterations author may come up to some instructional design framework most suitable for graphical constructs of that particular topic.

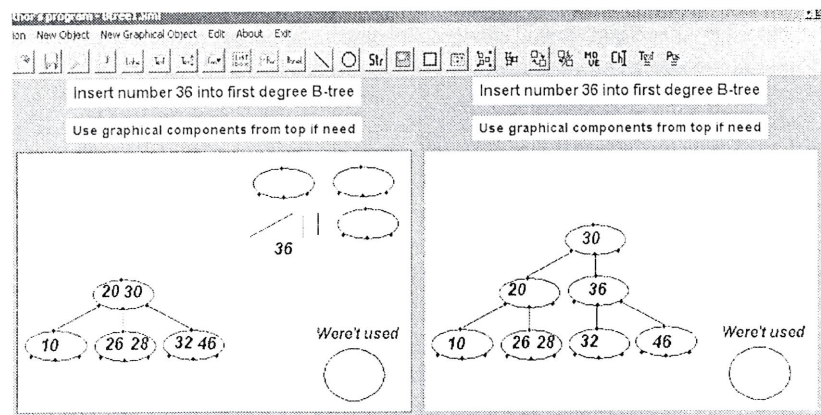


Fig. 7. B-tree task instructional design.

5. TestTool Implementation and Use

The TestTool is implemented as client server application using Java and web database (Apache+MySQL+PHP) technology. For standard networked configuration a client needs only browser to access web server and being logged in gets access to designated resources. The authoring subsystem starts every time if author logs in, whereas learner gets access to the tests or exam assigned for the group, to which that particular learner belongs. Both authoring and testing subsystems are designed as interactive Java applications, which in this networked configuration use question variants, questions, tests and exams stored in server's database. Administrator subsystem is web database application programmed using PHP as server-side scripting language.

The TestTool was designed and implemented at the technological higher education establishment. For an obvious reason from the beginning it was meant as a web-based medium to test and assess future engineers' knowledge and real problem understanding by looking at their capabilities to solve rather simple constructive tasks typical for one or another subject. Naturally the first systematic use of the TestTool in practice was carried out at Informatics Faculty of the Kaunas University of Technology. It was a "Data Structures" course module delivered for Computer Science undergraduates. Self-testing and final assessments have been carried out during two years in the fall and spring semesters having in total more than 450 undergraduate students. They demonstrated a high level of motivation and willingness in taking self-testing sessions, and also contributed to improvement of this kind of learning material.

TestTool gains more popularity among high school teachers too. It is mainly because Informatics faculty provides two years ICT course for high school teachers who have technical background. The curricula of the course include e-learning related modules and teachers are able to practice with TestTool implementing testing and assessment materials for their own subjects being taught in a school.

Though schools computerization in the country goes on rapidly, but there are still many schools where teachers do not have access to server based networked facilities and are unable to experiment and use the TestTool in full power. In order to facilitate adoption of the TestTool by teachers in those places the autonomous authoring and testing subsystems have been implemented. These Win32 applications enable authors to create, debug and store question variants using only local PC resources. Ready to use question variants may be uploaded to server database at any convenient time.

On the other hand, more wide use of the TestTool becomes feasible owing to rapidly growing number of networked classes in Lithuanian schools. A few initiatives for using TestTool in teaching practice have already grown up into distinct projects. For instance, the agreement between the School of Junior Physicist in Šiauliai city and Informatics Faculty has been recently signed aiming at cooperation in creation of testing and assessment materials for physics module. There are several other schools in Kaunas, Kaišiadorys where individual still experimental projects are carried out.

6. Conclusion

Our work on instructional designs along with experiments in using TestTool in real environment within higher education establishment forces a re-evaluation of what and how is being asked of learners in a traditional computer aided assessment systems. In this paper we emphasize an importance of a 'good' question type, which would allow computerized assessment of skills and real problem understanding. The graphical construct question type, which accommodates constructivist learner-centered style of learning, might be considered as our attempt to approach this complicated task. In general, on the ground of our two years practice of using the graphical construct for learning and assessment we argue that the power of this type of question estimated as an ability to create cognitively authentic task situation along with constructive way of task solution is advantageous and outweigh the limitations of the method being used to define the correctness of the answer. Further efforts in development and using of the TestTool are aimed at improvement of both methodological and technological basis for creation of meaningful assessment materials.

Acknowledgements

Authors give many thanks to MSc students Emilis Stuopys and Giedrius Paulikas for their studious programming efforts implementing TestTool.

References

- Anderson, L., D. Krathwohl (2001). *A Taxonomy for Learning, Teaching and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*. Longman, New York.
- Baniulis, K., B. Ciapas and V. Reklaitis (2000). TestTool: web-based automated tracing and grading system. In *The Proceedings of 3th International Workshop ICL2000*. Kassel University Press, Villach, pp. 1-7.
- Baniulis, K., V. Reklaitis and E. Stuopys (2001). Interactive multimedia authoring tool for Web-based testing. In *The Proceedings of World Conference ED-MEDIA 2001*. Tampere, pp. 84-85.
- Boyle, A.P., D.J. Edwards, D.M. Mackenzie, B. Mills, D. O'Hare, E.C. Morris, C.R.C. Paul, H. Wilkins and D.W. Williams (2000). Developments in on-line assessment – experiences and evaluation of using TRIADS and its potential for assessment in Electrical Engineering. *International Journal of Electrical Engineering Education*, 37(1), 74-85.
- Crofts, A., M. Foster, M. McAlpine, R. Rist and N. Tomes (2000). User experiences of choosing and implementing Web-based computer aided assessment. *Fourth International Computer Assisted Assessment Conference*, Loughborough. <http://www.loughborough.uk/service/fli/flicaa/conf2000/pdfs/contents.pdf> (accessed July, 2002).
- Davies, P. (2001). Computer aided assessment MUST be more than multiple-choice tests for it to be academically credible? *Fourth International Computer Assisted Assessment Conference*, Loughborough. <http://www.lboro.ac.uk/service/ltd/flicaa/conf2001/pdfs/e2.pdf> (accessed July, 2002).
- Mackenzie, D. (2000). Production and delivery of TRIADS Assessments on a university-wide basis. *Fourth International Computer Assisted Assessment Conference*, Loughborough. <http://www.lboro.ac.uk/service/ltd/flicaa/conf2000/pdfs/mackenzied.pdf> (accessed July, 2002).

- Macromedia Flash Documentation (2002). <http://www.macromedia.com/software/flash/> (accessed July 2002).
- Questionmark Perception Product Information (2002). <http://www.qmark.com/perception> (accessed July, 2002).
- Squires, D., and J. Preece (1999). Predicting quality in educational software: evaluating for learning, usability and synergy between them. *Interacting with Computers*, 11(5), 467-483.
- Standards for Quality On-Line Courses* (2002). <http://www.ideos.net/standards/id/> (accessed July, 2002).
- Stephens, D., and J. Mascia (1997). *Results of a Survey into the Use of Computer-Assisted Assessment in Institutions of Higher Education in UK*. <http://www.lboro.ac.uk/service/ltd/flicaa/downloads/survey.pdf> (accessed July, 2002).
- TRIADS – Assessment of Learning Outcomes (2002). <http://www.pcweb.liv.ac.uk/apboyle/triads/index.html> (accessed July, 2002).
- Vantaggiato, A. (1998). Automatic Exams and Course Administration on the Web. In *Proceedings of WebNet 98 – World Conf. of the WWW, Internet & Intranet*, vol. 2. Orlando, pp. 932-938.

K. Baniulis has graduated Computer sciences from Kaunas Polytechnic Institute. He received Doctor of technical sciences (PhD) degree from Kaunas Polytechnic Institute in 1971. He is an associated professor at Kaunas Technological University Computer Networking Department. His research interests are Information Technologies in Education, Multimedia and Artificial intelligence applications in e-learning.

V. Reklaitis has been involved in networking development at Kaunas University of Technology over the last 12 years. He headed Joint projects funded by EC under TEMPUS and INCO-Copernicus programmes related to both teaching of networking and developing of e-learning applications. Since April 2002 he is a professor at Graduate School of Information Systems, University of Electro-Communications, Tokyo.

Nuotolinio žinių vertinimo, testavimo ir mokymosi aplinka TestTool

Kazys BANIULIS, Vytautas RĖKLAITIS

Šiame straipsnyje mes nagrinėjame testavimo ir žinių vertinimo galimybes, kai norima įgyvendinti aukštesnius mokymosi tikslus. Sistema TestTool buvo projektuojama technologiniame universitete, siekiant sukurti nuotolinio testavimo ir žinių vertinimo aplinką, kurioje būsimieji inžinieriai spręstų konstravimo uždavinius. Šiam tikslui tradicinė testavimo sistema su pasirenkamais atsakymais netiko. Todėl mes įvedėme grafinio tipo klausimus, kurie akumuliuotų konstruktivistinį mokymosi principą, orientuotą į individualų mokinių. Straipsnyje aptariame, kaip grafinio tipo klausimas išplečia TestTool funkcines galimybes kompiuterizuoti Duomenų Struktūrų modulį, skaitomą informatikos ir kitų specialybių studentams. Straipsnio pabaigoje išnaginėjame TestTool platesnio įdiegimo galimybes.