Towards Intelligence and Flexibility of Learning and Knowledge Testing Environments

Nerijus AUKSTAKALNIS,

Kazys BANIULIS

and

Bronius TAMULYNAS
Computer Networking Dept.,
Kaunas University of Technology,
Studentu 50, 51368 Kaunas, Lithuania

ABSTRACT

The proposed goal oriented knowledge acquisition and assessment are based on the flexible educational model and allows to implement an adaptive control of the enhanced learning process according to the requirements of student's knowledge level, his state of cognition and subject learning history. The enhanced learner knowledge model specifies how the cognition state of the user will be achieved step by step. The use case actions definition is a starting point of the specification, which depends on different levels of learning scenarios and user cognition sub goals. The use case actions specification is used as a basis to set the requirements for service software specification and attributes of learning objects respectively. The paper presents the enhanced architecture of the student self-evaluation and on-line assessment system TestTool. The system is explored as an assessment engine capable of supporting and improving the individualized intelligent goal oriented self-instructional and simulation based mode of learning, grounded on the GRID distributed service architecture.

Keywords: assessment, learning, flexible educational model, collaboration, domain of knowledge.

1. INTRODUCTION

In current practices, most of the e-learning solutions available on the market support only a specific aspect of the learning process: the content delivery (content centered approach) based mainly on the information transfer paradigm. This paradigm focuses on the content and the key authoritative figure of a teacher who provides information. In order to stimulate an effective learning process it is necessary to individuate a suitable technological infrastructure. In the innovative vision the learning process within a learning environment can be effective only using an approach, which takes into account some fundamental characteristics of learning activity and learning within a contextualized environment [1, 2]. Another fundamental and innovative aspect is the possibility of personalizing the learning process with respect to the needs of each learner. We will investigate strategies and methods to determine in the first place the learning style preferred by the student, then suitable methodologies which allow performing a personalized educational process on this basis. So, a self-adapting system can be defined as the capability and intelligence of the learning environment, which can be increased by using a suitable and innovative domain of knowledge and knowledge state of the learner [1, 2 and 3]. The goal-oriented approach aspires to promote and support a new paradigm focused on knowledge construction using experiential and collaborative learning approaches in a contextualized, personalized way.

Considering humans at the centre, learning is clearly a social, constructive phenomenon. It occurs as a side effect of realistic simulations, interactions, conversations, collaborations and enhanced presence in dynamic Virtual Communities [7]. This new paradigm is based on a learner centered approach by creating and adapting learning paths according to learner's previous knowledge, preferences, skills, and the preferred learning style in collaboration with other students, teachers, tutors, or experts. Rather than stressing the memorization of information, learning activities are aimed at assisting the learner in the construction of an autonomous, functional base of knowledge and skills.

The innovative research on the GRID technologies shows the right way for achieving an effective learning and at the moment it seems to be the technological infrastructure that fits such requirements best [4, 5 and 6]. Indeed they allow to access and integrate different technologies, resources and contents that are required in order to realize a new paradigm. It is the most promising approach to realize an infrastructure that will allow learning process actors to collaborate, to take part in realistic simulations, to use and share high quality learning data and to innovate solutions of learning and training. GRID will be able to support learning processes allowing each learner to use the resources already existing on-line in a transparent and collaborative manner by facilitating and managing dynamic conversations with other human and artificial actors available on the GRID. The definition and implementation of an advanced service-oriented GRID based software architecture for learning is driven by the pedagogical needs and by the requirements provided by the test-beds. We are trying to realize this conception by using TestTool (TT) knowledge testing environment.

2. USE CASE GOAL ORIENTED LEARNING

When the learner enters the simulated environment, he finds himself in some initial state and his objective is to move to some final (solution) state by performing appropriate actions, operations and moves. From the cognitive perspective, when the learner tackles a new problem he uses a number of cognitive strategies that involve inductive and deductive reasoning, learning by trials, errors and insight. The user performs a sequence of (either mental or concrete) operations that allow him to move from one to another knowledge state until the final state (the solution of the problem) is reached. So, in the simplest terms this principle may be defined as mapping from the knowledge state of a student to the domain of knowledge space of the problem [1, 7].

Domain of knowledge
Domain of knowledge is a collection of items (e.g., theories, examples, problems, questions, exercises, models and other learning objects (LO)) in a given field of knowledge. *The knowledge state of a learner* is a set of all items this student actually masters. In the domain of knowledge the items are linked by surmise relation, which allows identifying the prerequisites for each item. This relation is a partial order for representing implication relations among items in a given domain of knowledge. The knowledge state of the learner is mapped into a latent skill state and the mapping itself is called skills map. The knowledge space is just a model of the cognitive organization of some learning material. As such to become a valid representation with respect to some existing population of learners, it has to be tested empirically. The learner knowledge model [2, 3, and 7] can be constructed as a sub graph of the global knowledge domain graph. Such knowledge representation model belongs to the most general theory concerning multi-graphs. The domain knowledge model can be based on graph paradigm by decomposing information into atomic units and finding the connections among the units themselves: motivational or historical type, difficulty degrees. Therefore the structure consists of both the information units and their links.

The goal oriented knowledge based learning
Functionally the general knowledge acquisition process can be divided into three categories of learning goals:
- ✓ To know the theory and methods;
- ✓ To know the theory and methods and be able to use them in standard situations;
- ✓ To know the theory and methods and be able to use them in unknown situations.

Each category can be decomposed into several components and described as a sequence of sub goals. Naturally these sub goals are defined in relation to prerequisites given by the domain of knowledge. This means that it implies the use case specification process grounded on an efficient functional learning requirements specification with a strong focus on pedagogy driven requirements. Thus the first category can be (less or more) considered as content centered approach. The second category may imply some specific problem oriented simulation models. The last one includes more complex simulation models, which enable to create more purposeful space of actions.

The goal-oriented learning (GOL) context has to answer the question: what observable result of value is obtained by the tutor (e-learning service designer) and the learner (e-learning service user), because any e-learning scenario has two actors. The main learning goal of the use case has to be specified in natural language as a main functional requirement. The use case specification is a starting point to specify how the goal will be achieved and should include different learner's knowledge state levels (sub goals).

David Kolb describes learning in general as a four-step process [8]: (1) watching, (2) thinking, (3) feeling, and (4) doing. Learners have concrete experiences that allow them to reflect on new experiences from different perspectives. From reflective observations, learners engage in abstract conceptualization, creating generalizations or principles that integrate observations into sound theories. Finally, learners use these generalizations or theories as guides to further action. The active experimentation allows learners to test what they have learned in new, more complex situations. The result is another concrete experience, but this time at a more complex level. Experiential learning can be defined therefore in terms of a learning model, which begins with the experience followed by reflection, discussion, analysis and evaluation of the experience. The assumption is that individuals seldom learn from the experience unless they assess it, assigning their own meaning in terms of their own goals, aims, ambitions and expectations.

The knowledge acquisition through experiential learning actions depends on the type of the main goal and simulation-based learning environment. According to the learning goal category the ultimate form of simulation via intelligent tutoring systems must imply: 1) instructional content centered models 2) simulation tools that allow perceiving specific instructions of content understanding 3) a possibility to pretend the use of real applications or extended models. Besides, an intelligent tutoring environment simulates the content and process of learning and support feedback mechanisms to correct student errors.

3. SIMULATION-BASED KNOWLEDGE ACQUISITION

Today high-end simulations are most often hand-built masterpieces. They require lots of story boarding, if-then rules development and in many cases lots of video footage shooting or graphic modeling to come up with what will be an interactive and engaging simulation.

A simulation engine is a system that will eventually mean that every scenario has to be scripted. Modeling software, databases and rules will enable powerful computing platforms to mimic various knowledge acquisition scenarios. All this will be costly and in the short term it means that more effective tools will allow more simulations to be developed for the same amount of money.

Simulations have been a great promise in education and in training for a long time. There is an intuitive feeling that people will learn much from a compelling, interactive experience with the scientific process one is trying to understand or the complex device one is trying to master. However, studies on the value of simulation have yielded mixed results. Kulick [9] reports that the use of simulation only found modest learning effects or could not find any increase in learning at all. Shute and her colleagues found that while some trainee populations thrive in such environments others flounder [10, 11].

Traditionally it has been difficult to embed a valid assessment and feedback in complex simulations. The evidence-centered design (ECD) is a method that can address these problems and enable the development of the robust and valid simulation-based assessment and training systems [12]. ECD enables to develop valid profiles of user's skills and abilities from the rich behavioral sequences that human interactions with complex simulations typically produce.

One way of assuring that simulations are valuable is to include the assessment as a part of the simulation. If we can accurately measure what people have learned as a result of using a simulation, the value of the simulation is much clearer. It depends on the type and form of learning actions implemented, learning goals to be achieved and the way in which learning results could be evaluated. Standards of LO are oriented to support implementation links with learner's experiential achievements as well [13].

One might give students or trainees a traditional test after completing a training experience to see what they have learned. Nevertheless, the traditional knowledge testing is not a suitable instrument for assessing this type of skills because: a) the sequence of individual actions through the interaction with the simulation model is tied to the context; b) a traditional testing questionnaire is designed for one particular piece of knowledge (one question – one fact); c) typically we want to assess a whole constellation of skills and abilities from the evidence coming from the students' interaction with a simulation, methods for analyzing the sequence of behaviors to infer these abilities are not as obvious i.e. one simulation – constellation of skills and abilities. In this case, [12] the simulation environment itself may be the best place to assess these skills. During the simulation students produce rich sequences of actions while performing complex tasks. They draw upon the very skills we want to assess. The evidence needed to assess the skills is provided by the students' interactions with the simulation itself.

A specific simulation engine can be implemented using authorized graphical environments. In this case the more complex and flexible knowledge acquisition scenarios and skills assessment strategies enable to achieve well-balanced educational goals. Such possibilities have TRIADS [14] and TestTool [6, 7] systems. The main peculiarity of the TT simulation engine is the compatibility of the experiential and knowledge assessment phases using a specific graphical environment for both cases.

TestTool graphical simulation environment (GSE)
TestTool GSE consists of two parts: the authoring graphical simulation environment for the teacher and the simulation environment for the learner. The teacher authoring GSE is used to design knowledge acquisition scenarios as a basis of the interactive simulation model. The knowledge acquisition scenario implies assignments, initial data and admissible answers. The authoring GSE itself consist of several tools for drawing objects, the collection of simulation control elements, multimedia and sound effects. All these tools are created using JAVA script technology. The main learning goal of the GSE is to provide large-scale possibilities and actions to search the solutions for a set of selected situations as well as to assess the quality of student's answers respectively.

Simulation-based learning scenarios: demonstration, self-testing and knowledge assessment
Using the simulation-based learning demonstration mode the student can scan typical situations and analyze sequences of relevant actions. In this case the simulation engine enables learners to use an effective way for a simulation behavior of complex systems and to study rules and dynamics of the model: to make changes of many different parameters and to see results of their own actions immediately. Learners can tweak and tune their answers as much as they want.

The self-testing or training scenario allows solving the situations while the right answers are achieved. After a successful training the knowledge assessment can be started.

The process of skills and knowledge assessment implies solution search actions for several randomly presented situations. After a particular solution is made the system compares the answer and the result is sent to the knowledge assessment database. The learner can check his answers scores in the summary table.

The teacher designs simulation-based knowledge acquisition and testing scenarios: he defines all simulation actions, modes, testing schedule, duration, variety and sequences of tasks. Knowledge acquisition testing tasks could be performed just once. The teacher makes the final decision to allow repeating the test or not.

The simulation-based knowledge acquisition and testing allows implementing specific experiential learning strategies:
- ✓ The decomposition of the knowledge domain into atomic units of skills;
- ✓ The use of different strategies for more effective knowledge acquisition;
- ✓ The variety and complexity of actions;
- ✓ The variety of learning goals according to the Bloom's taxonomy [15].

Simulation-based teaching strategies
The design of simulation-based scenarios must be oriented to embody specific didactic solutions: the selection of specific learning areas for the skills representation (decision making rules, definitions and etc.) and the collection of relevant examples to illustrate the phenomena to be studied.

We use the following teaching strategy to form abstract concepts: showing a number of examples and non-examples of idea so that learners would notice similarities and differences among these examples and then would grasp the "rule", discovery learning with an increased self-study effort, learning by problem solving with an increased feedback effort. The design and selection of representative situations have a great and important impact to search for a relevant sequence of actions. The simulation-based training model according to the teacher-authors scenario and demo examples enables the learner to understand the main features of the teacher's style.

Interactivity
Very important aspects of simulation-based learning are directly related to the interactivity levels of student actions as well as the simulation model complexity to be used for achieving higher didactic learning goals. According to e-Learning standards [13] the level of learning interactivity can be very low, low, medium, high, and very high. Using the GSE we can create and support LO with high interactivity (a lot of simulation actions control elements) or low interactivity (the sequence of instructional actions).

For example, the simulation of actions with dynamic data lists implies several fragments of source program codes or specific mimic of data manipulation at the end of the list. In the case when the student must recognize only the kind of list that the source program code fragment represents - linear, cyclic, double oriented or nested – the interactivity can be lower. On the other hand a higher level of interactivity is used when the program source code implies several mistakes or the student must give a detail evaluation of more complicated source code sentences. The next step of simulation complexity may imply the tasks to update the source code, to trace the given algorithm or to correct the unfinished source code part as well.

Personalization and flexibility of GSE
The flexibility of GSE enables to extend the experiential knowledge acquisition applications for a rather large scale of

subjects with different complexity of knowledge domains (specific explicit themes or ill-defined phenomena). A preliminary expertise with Data Structure studies module approves such simulation practice and encourages applying it as a high-motivated part of the goal-oriented learning.

Such a way of organizing training environments enables to track the improvements of the user along the whole training process. This implies a dynamic adaptation of the system to the user's skills and performance (personalization) so that his motivation and mental activity remains at a rational level during the whole training session [7]. The mechanism at the basis of this adaptation is the performance and skill assessment and monitoring. The skill is the capability to perform a given sequence of operations in a purposeful way and the meta-skill is the ability to combine such sequences. Such considerations suggest that the tool sharing among all the users is the key element to strengthen the effectiveness of the learning process in the situated learning framework. So all the learners everywhere and at any time are provided with the same equipment that enables to know and be able to take part actively in the common knowledge building.

Finally, it is important for the students to use particular resources, which, due to their specific character, can be present only on remote systems. In this case the learner can invoke any application from a remote system, use the system best suited for executing that particular application, access data securely and consistently from remote sites, exploit multiple systems to complete complex tasks in an economical manner, or use multiple systems to solve large problems that exceed the capacity of a single one.

4. LEARNING SCENARIOS AND THE USE CASE SPECIFICATION OF ACTIONS

In regard to the learning requirements an efficient functional specification has a strong focus on pedagogy driven requirements. This means for us that we need to define our Use Cases specification process and have to think about software requirements respectively. If so, then it would be unforgivable and unjustifiable to ignore the best software engineering practices. Particularly our 'strong focus on pedagogy' associates with writing requirements in context where *a difference between thinking* of system requirements in terms of a list of features or functions and thinking of features and functions in *a goal-oriented context* is particularly emphasized [17]. Proceeding with the goal-oriented context the use case specification has to answer the question: what observable result of the value is obtained by the e-learning service designer and by the e-learning service user, because any e-learning scenario has two actors. Then two complementary specifications are needed. In both use-cases specification forms the solution of the main goal-orientation is not explicit so far. We think the main goal of the use-case has to be specified in natural language as a main functional requirement (learning goal!). The use-case specification is a starting point to specify how the goal is going to be achieved (the flow of events) (Table 1).

The issue of evaluating whether the goal has been achieved or not is very important for the e-learning service (or application) and untypical if to compare with conventional software engineering.

Software Engineering uses mainly two groups of Software Process models: one like UP (very bureaucratic) and another of the agile processes group like XP (less bureaucratic), which manifests a faster start of coding. In connection with the latter the following question is relevant: how can the learner's action

specification model help us to build an executable code or to define a search procedure of the relevant LO in the LO repository, respectively. Meanwhile, there is no such a concept as use-cases levels in Software Engineering. We could use component or deployment diagrams along with high-level requirements.

Table 1. Learner's actions specification using TT GSE

<Name>	Experiential and active learning
General description related LO	Several simulation models are created for a specific theme. Learners can observe and analyze their own actions in the context of the active learning. They can achieve a specific goal by working with the model in GSE to perceive particular decision rules
Primary Actor(s)	Learner, student, course master
Primary Goal	To master methods and particular actions using TT and models of GSE to learn them in standard situations
Other users may be involved	Course manager, student's collaboration partners
Preconditions	Specific knowledge of the theory to be studied and general basic simulation rules with GSE
Main success scenario	1. Login of the user 2. Choose: the regime … simulation model … 3.Initialization of parameters 4. New regime or model ….. 5. Assessment of knowledge 6. Log off
Extensions	2a. Learner requires additional information 4a. Choose a new topic 5a. New model or list of new situations
Success (post conditions)	Learner get positive evaluation and can be allowed to start the next learning session
Alternative flows	If failed -> to return to the theory or methods study session
Special requirements	If needed

5. CASE STUDY: STUDENT AND DATA STRUCTURE COURSE MODELS

Case study is based on the use of the graphical assessment system TestTool [5, 7] in the Data structure study module.

Creation of the course model
The course model can be defined as a collection of items in a given field of knowledge [16]. The course structure is created according to learner's needs, aims and objectives. A hierarchical principle is used to organize the learning context, i.e. the context can be enumerated or rendered as a graph by relational topics, which are studied in any order. Various ways of analysis and graphical rendering are used to aggregate the structure of the course. One of many possible forms is the concept map. It should begin with the name of the key topic or concept and link it to a number of related concepts. Then the

names of concepts related to these topics should be connected as well. A different technique for the content presentation is used: hierarchical lists, chronological lists, content matrices, causal schemes, etc.

Case study
Teachers-experts define the three levels of knowledge (basic, intermediate and expert) the learners should be familiar with. These definitions are used to develop a course model and to form student model subsets. Creating models [2,7] the criteria of educational goals according to Bloom's taxonomy are used. The course model creation is simple:
- ✓ The teacher-expert creates learning objects and describes their attributes;
- ✓ The teacher-expert defines the relations between one LO and several others LOs possessing some specific relationships;
- ✓ Several different relationships can existed between two LO;
- ✓ The possible relationships are defined initially or added/modified later.
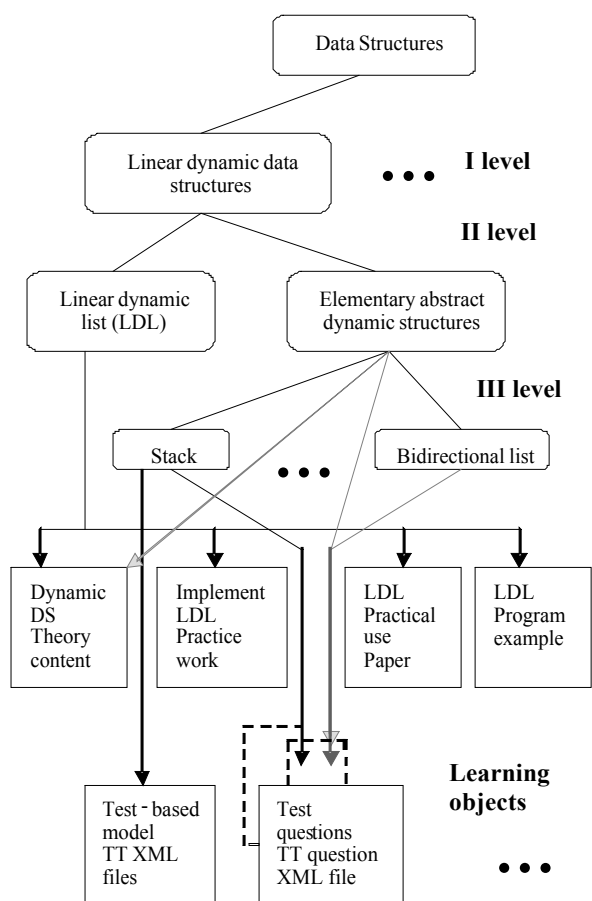

Figure 1. Data structure course model

According to the existing e-learning standards attributes and relationships (LO metadata) are described in XML [7]. TestTool test-exercises (LO data) are described in the same way.

In order to realize the first domain of knowledge the main arguments for the TestTool basis are:
- ✓ The environment is authorized and is easy modifiable according to the needs;

- ✓ XML files are convenient for the LO presentation by adding several new features.

The domain of knowledge as the data structure course and student models, as a subset of the Course model are defined in [7]. The student model is defined according to various criteria subsets of Course Semantic Network. Graph analysis methods take into account learner's possibilities and his learning progress.

TestTool and GRID Service Architecture
Based on the integrity and interoperability of distributed learning object systems, the experimental TestTool version for GRID was implemented [7]. In order to transform the existing TestTool system to the GRIDTT version the following steps are needed:
1. The TestTool system based on Web services is created as a distributed system. The TestTool system consists of three large subsystems of services: Client, LO Repository closely linked with LO Registry, Learner Repository. Client is a general subsystem, which realizes interfaces for TT users. LO repository is the service for storage, registration and retrieval of LOs. Learners' repository is a learners' database containing individual student data records and testing results. The typical service sequence for a learner is:
- A student logs in through Learner Repository and chooses an eligible test from the list;
- LO Registry is searched for the chosen test, LO Repository address and LO identification data are retrieved;
- Learner Repository calls LO Factory and creates instant aggregated questions;
- A learner communicates with the visualized LO to answer the questions;
- LO sends the answers and the evaluations to the Learner Repository.

2. To fulfill knowledge domain requirements Course model and Learner model solutions are performed (Figure 1).

There are a lot of additional tools in the existing integrated systems. The creation of a Web service based e-learning system produce new challenges, the course administration becomes more complicated due to a more complex architecture. The administrator has to aggregate the course from learning objects distributed in various places. The course aggregation becomes complicated for an inexperienced user and the possibility that something will be missed or mistakes will be made is high. One of the main goals of the distributed system (learning GRID) is to create and improve services for increasing the level of automation and users utility.

A high level visualization for the course administration reduces the interest in the structure of the course itself and the administrator has to specify LO and relations between them. The graphical course aggregation tool or course visualization using tree structures in the HTML file are no useful. LO is then located by the system of queries according to the attributes the administrator has specified.

The installation of additional software for the system to work irritates users when they have to do it themselves. To solve such problem technologies supporting standard operating system should be used. Currently, a new more intelligent version of TT based on Web services is being created. The content of LO created using the earlier version of TT *Author* program is reusable in the new version. Reusability is achieved by using the XML converter in order to transfer questions and tests into a new format. The XML converter can be used to implement standardization of any learning objects as well.

6. CONCLUSIONS AND DISCUSSIONS

The enhanced learner knowledge model specifies how the cognition states of the user will be achieved step by step. The use case actions definition is a starting point of the specification, which depends on different levels of learning scenarios and user cognition sub goals. The use case actions specification is used as a basis to set requirements for the service software specification and attributes of learning objects respectively. The paper presents the enhanced architecture of the student self-evaluation and on-line assessment system TestTool. The system is explored as an assessment engine capable of supporting and improving the individualized intelligent goal oriented self-instructional and simulation based mode of learning, grounded on the GRID distributed service architecture. The research consists of two parts: the conceptualization of goal-oriented knowledge based learning environments and the development of the use case specification of actions using the intelligent TestTool version. Web services based TT version of the graphical assessment system is used already for four years for the Data structure study module (400 students per year). Currently, we are preparing knowledge testing package for introductory course "Computer sciences and programming" (4000 students). TT learning and knowledge testing environment will be adapted as a tutorial for mobile phone operators as well as to certificate the members of Lithuanian hunting association.

7. REFERENCES

[1] N. Capuano, A. Gaeta, G. Laria, F. Orciuoli, P.Ritrovato, "How to use technology for building next generation learning environments", **Proc. of the 2-th International LeGE-WG Workshop: Towards a European Learning GRID Infrastructure,** Paris, France, 2003.

[2] K. Baniulis, B. Tamulynas, "Intelligent support of web-based knowledge assessment system", **IEEE International Conference on Advanced Learning Technologies Proceedings,** Kazan, Tatarstan, Russia, IEEE Learning Technology, 9-12 September 2002, pp. 551-552.

[3] K. Baniulis, B. Tamulynas, "Flexible Learning in an Intelligent Tutoring Environment", **New Media and Telematic Technologies for Education in Eastern European Countries,** Twente University Press, 1997, pp. 395-409.

[4] V. Reklaitis, K. Baniulis, T. Okamoto, "Shaping e_Learning applications for a service oriented GRID", **2nd International LeGE-W G W o rkshop on e-Learning and Grid Technologies: A Fundamental Challenge for Europe,** Paris, France. 3rd & 4th March 2003.

[5] K. Baniulis, V. Reklaitis, TestTool: "Web-based Testing, Assessment", *Learning, Informatics in Education, ISSN 1648-5831,* **1**, Vilnius, Institute of Mathematics and Informatics, 2002, pp. 17-30.

[6] V. Reklaitis, K. Baniulis, G. Paulikas, "Implementation of Assessment Web-service for TestTool", **Proc. of the International Conference: Advanced learning technologies and applications (ALTA'03),** Kaunas, Lithuania, 2003, pp. 39-40.

[7] K. Baniulis, B. Tamulynas, N. Aukstakalnis, "Case Study of Virtual Organization Learning and Knowledge Testing Environments", **Proc. of the 3rd International LeGE-WG Workshop: GRID Infrastructure to Support Future Technology Enhanced Learning,** Berlin, Germany, 3 December 2003.

[8] D. Kolb, **Experiential learning: experience as the source of learning and development.** Englewood Clifs, NJ: Prentice-Hall, 1984.

[9] A.J. Kulick, "School Mathematics and science program benefit from instructional technology", **InfoBrief**, NSF-03-301, 2002.

[10] V.J. Shute, R. Glaser, "Large-scale evaluation of an intelligent tutoring system", **Interactive Learning Environments,** Smithtown, 1990, pp.51-76.

[11] V.J. Shute, R. Glaser, K. Raghavan, "Inference and discovery in an exploratory laboratory", In P.L. Ackerman, R.J. Sternberg & R. Glaser (Eds.), **Learning and Individual Differences,** New York, NY: W.H. Freeman, 1989, pp. 279-326.

[12] M. Bauer, D. Williamson, R. Mislevy, J. Behrens, "Using Evidence-Centered Design to Develop Advanced Simulation-Based Assessment and Training" **World Conference on E-Learning in Corp., Govt., Health., & Higher Ed,** 2003, pp. 1495-1502.

[13] IEEE LOM. Draft Standard for Learning Object Metadata, **http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf,** 2002.

[14] Don Mackenzie, "Recent Developments in the Tripartite Interactive Assessment Delivery System (TRIADS)", **University of Derby, http://www.derby.ac.uk/ciad/lough99pr.html.**

[15] L. Anderson, D. Krathwohl, **A Taxonomy for Learning, Teaching and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives,** Longman, New York, 2001.

[16] L. Stefanutti, D. Albert, C. Hockemeyer, " Derivation of Knowledge structures for distributed learning objects", **Proc. of the 3rd International LeGE-WG Workshop: GRID Infrastructure to Support Future Technology Enhanced Learning,** Berlin, Germany, 3 December, 2003.

[17] C. Larman, **Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development,** Prentice-Hall, 2000 ed., http://www.craiglarman.com/, ISBN 0130925691.