Education Cluster for Intelligent Provision of eLearning Services

Stanimir Stoyanov
Department of Computer Systems
University of Plovidiv "Paisii Hilendarski",
Plovdiv, Bulgaria
stani@uni-plovdiv.bg

Emil Doychev
Department of Computer Systems
University of Plovidiv "Paisii Hilendarski",
Plovdiv, Bulgaria
e.doychev@isy-dc.com

Veselina Valkanova
Department of Computer Systems
University of Plovidiv "Paisii Hilendarski",
Plovdiv, Bulgaria
veselinaviva9@gmail.com

Georgi Cholakov
Department of Computer Systems
University of Plovidiv "Paisii Hilendarski",
Plovdiv, Bulgaria
gcholakov@uni-plovdiv.bg

***Abstract* - The DeLC (Distributed eLearning Center) project aims to develop an interactive, proactive and personalized e-Learning environment. Two nodes of DeLC architecture are presented in the paper. The first node is a service-oriented portal, providing personalized educational services and teaching material. The second node is an agent-oriented server which incorporates three agents supporting electronic testing of students. This paper focuses on the hybrid architecture, known as education cluster, which integrates the two nodes. The main results of the use of the cluster are increased interest and more active inclusion of the students in the education process.**

Keywords – service-oriented architectures; intelligent agents; e-learning.

I. Introduction

Recently, the interest towards electronic education is growing. As a result, many universities developed and implemented their own systems for electronic and long-distance education. Alternatively, many of the large IT corporations (e.g., Microsoft Class Server [1], IBM [2], and HP [3]) developed commercial systems. On the other hand, there are different open-source systems available on the market (the best known is Moodle [4]). A number of standards for electronic and life-long learning are also emerging. There are many organizations working to develop specifications and standards such as Innovation Information Management System Global Learning Consortium [5], Advance Distributed Learning [6], IEEE, ISO, etc., to provide a framework for e-Learning architectures, to facilitate interoperability, content packaging, content management, Learning Object Meta data, course sequencing, and others. One significant example is played by the Sharable Content Object Reference Model 2004 (SCORM 2004) standard [7].

DeLC (Distributed eLearning Center) [8, 9] is one of the projects aiming to develop an environment that supports electronic and long-distance forms of education. Why, despite the presence of so many systems, do we find it necessary to dwell on this subject? DeLC tries to provide architectural support for more effective eLearning systems which involve the students in a more personalized and creative education process and which stimulate students' activities and cooperation. An important goal of this project is the development and experimentation with prototypes of such architectures that are service-and-agent-oriented. To achieve it, we developed an environment which provides teaching materials and educational electronic services.

Furthermore, in many of the existing e-Learning systems, the interaction with the teachers is somewhat static – it is achieved mainly through pre-defined templates for choosing information resources. The information resources are the electronic equivalent of the traditional textbooks. Some of the existing systems use visualization and animation for improving the presentation of the teaching materials. In our project, we would like to research how such architectures can promote the development of electronic education environments, which support an interactive, reactive, proactive and personalized process of education and stimulate the students' creative and innovative thinking and performance.

The DeLC is a network, consisting of separate nodes, called eLearning Nodes. Nodes model real units (laboratories, departments, faculties, colleges, and universities), which offer a complete or partial educational cycle. Each eLearning Node is an autonomous host of a set of electronic services. This network configuration enables access, incorporation, use and integration of electronic services located on the different nodes. The eLearning Nodes can be isolated or integrated in more complex virtual structures, called clusters. Remote eService activation and integration is possible only within a given

 ISBN: 978-1-61208-185-4

cluster. In the network model, we can easily create new clusters, reorganize or remove existing ones. The reorganization is virtual and it does not affect the real organization. For example, the reorganization of an existing cluster can be made not by removing a node but only by denying access to the services it offers. The reorganization does not disturb other nodes' function as nodes are autonomous self-sufficient educational units providing one or more integral services. An important feature of the eLearning Nodes is the access to supported services and electronic content. In relation to the access there are two kinds of nodes – mobile access eLearning Node [10] and fixed access eLearning Node [11].

In this publication, we present the architecture of a standardized eLearning Node with fixed access, implemented as an education portal. Furthermore, an education cluster with hybrid (service-and-agent-oriented) architecture is described. Agents, which are resident in the cluster, are presented as well.

II. EDUCATION PORTAL

The fixed node is implemented as an education portal with a service-oriented and multi-layered architecture, consisting of three logical layers: user interface, e-services and digital libraries (Fig. 1).

The user interface supports the connection between the users and the portal. It allows users to register in the system and create their own personalized educational environment. The user interface provides visualization and access to services, depending on the user's role which is assigned during the registration.

The e-services are classified in two groups: engines and eLearning services. The engines are invisible for the users and their main purpose is to assist in processing eLearning services. The engines support and manage the meta-data in the portal. Using the provided data, they can effectively support the activation, execution and completion of the eLearning services. In the portal architecture, the following engines are incorporated: SCORM Engine, Test Engine, Event Engine, Integration & User Profiling Engine and AV-Call Processor.

SCORM Engine is an interpreter of the electronic content, developed in accordance with the SCORM 2004 standard.

The Test Engine assists in performing electronic testing through the portal. It processes the meta-objects, which describe the questions and patterns of the tests.

The Event Engine supports a model for event management, enabling the users to see and create events and be notified for them in advance. The events in the system reflect important moments for the users, such as a lecture, examination, test, national holiday, birthday, etc. Each event is characterized by attributes, such as a name, start and end date and time, details, and information whether it is a recurring one, as well as rules for its recurrence. The Event Engine supports yearly, monthly and weekly recurrence.

The Integration & User Profiling implements the supported user model.

eLearning services are grouped in three categories:

- Services for training, organizing and planning of the educational process;
- Services for conduction and management of the educational process – here belong services as electronic lectures, electronic testing, online and offline consultations;
- Services for recording and documenting the educational process – they support automated generation of documents recording the educational process in the form of examination protocols, students' books, teachers' personal notebooks and archives).

The third layer contains electronic content in the form of digital repositories (libraries). In the current version, digital libraries are implemented for electronic lecture courses, questions for knowledge testing, electronic tests templates, course projects and diploma theses. The portal services can work directly with the digital libraries. Moreover, a generalized catalog representing the libraries' contents, is provided to the users.

With the implementation of the portal we aim at providing personalized eLearning services and teaching material. A multi-aspect model of personalization is implemented in the architecture. The first aspect is user classification and role organization. The users' profiles can be classified by roles, user groups, communities, and organizations. The standard user profile includes the following attributes:

- Standard attributes – necessary for user identification through username, password, e-mail, and others;
- Extended attributes – addresses, phone numbers, Internet pages, IM, social networks' contacts, and others;
- DeLC custom attributes – other user identifications. For example, for users with role "student" these can be faculty number, subject, faculty, and course.

The portal gives an opportunity for extending the user profile with some additional attributes. The users' profiles contain the complete information needed for personalization of the portal services provided by DeLC, educational content and user interface. The profile is created automatically during the user's first login, through a request to the university's database, after providing the standard and custom attributes. The integration with the university database and other external components is supported by the Integration & User Profiling Engine. Extended attributes are provided by the user. During the following user's logins the information in their profile is synchronized, as newer updates in the university's database are automatically migrated in the user's profile, for example change of course or subject.

The second aspect is structuring of the teaching material, which is saved in the digital libraries. The teachers have the freedom to specify different structural schemes according to the desired teaching approach.

The third aspect is personalization of the provided eLearning services. In the current portal version it is possible to generate individual tests. The student can use personalized schedules and calendars and individual student reports and reviews can be prepared.

III. Education Cluster

In order to enhance the portal architecture so that it can provide eLearning services in a more intelligent manner, we are going to extend it with new intelligent components implemented as agents. In the same time we aim at a pro-active architecture. The pro-activity improves the system's usefulness and friendliness to the users. Pro-activity means that the software can operate "on behalf of the user" and "activate itself" when it "estimates" that its intervention is necessary. Pro-activity can be ensured through "reinforcement" of the portal architecture with intelligent components, which demonstrate proactive behavior. Two approaches are available:

- Direct integration of intelligent agents in the currently existing education portal;
- Building an education cluster according to the DeLC concept.

For different reasons the latter approach has been chosen where intelligent agents, called "assistants", will "live" in a newly-built agent-oriented server known as "Agent Village" (Fig. 2).

Finding solutions to the following three problems is important for the creation of an education cluster:

- What will the architecture of the Agent Village (AV) be?
- How will the communication between the service-oriented architecture of the portal and the agent-oriented architecture of the Agent Village be achieved?
- What about the particular assistants that will reside at the Agent Village?

A. Agent Village Architecture

The Agent Village has to provide an environment where the assistants can operate in an intelligent manner. "Intelligent" suggests that the agents can expose interactive, reactive, proactive and personalized behavior in respect to users' requests.

Different information resources, which are not saved in the digital libraries, will be located in the agents' environment. In conformity with DeLC's multi-aspect concept three models have to be implemented as ontologies, in order to ensure the next aspect of personalization in the education cluster – student model, pedagogical model an domain (discipline) model.

B. Portal-AV Interaction

The connection between the portal and the AV node is made through the middle layer of the portal architecture, where the electronic services are located. Depending on the direction of the assistance required we distinguish reactive and proactive behavior of the architecture.

In the reactive behavior, the interaction between the two nodes is initiated by the portal. This is necessary when a user request is processed and a service needs "expert" assistance. The service addresses the corresponding agent located in the AV. The problem is that the services are passive and static software modules in nature, intended

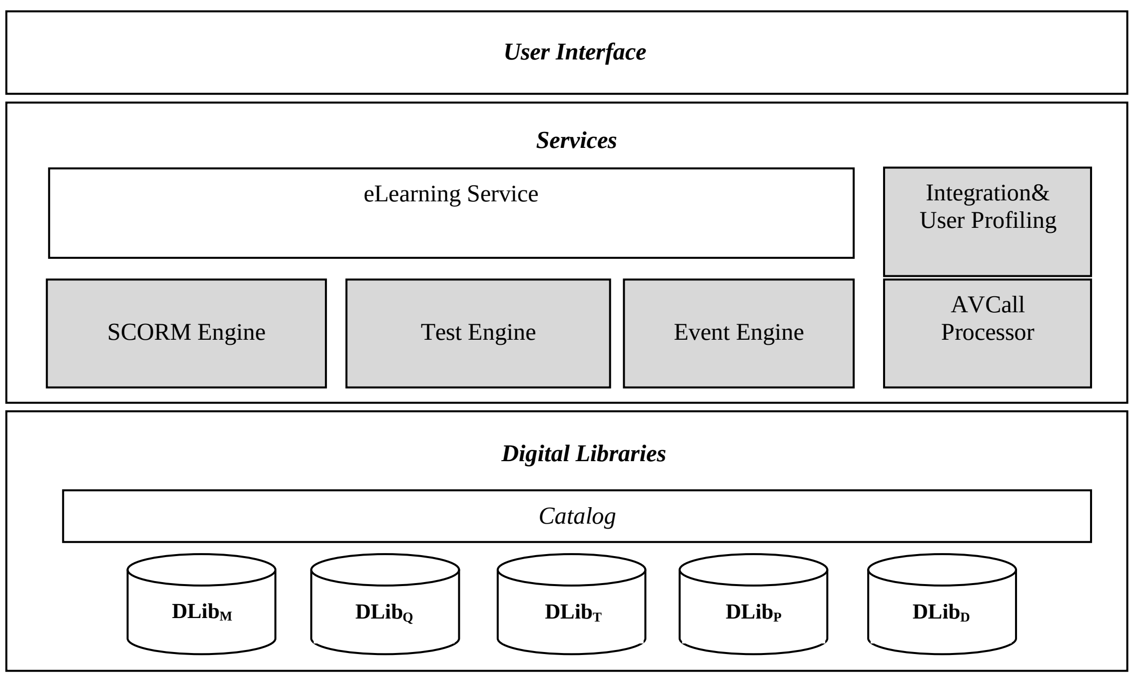

Figure 1. DeLC Education Portal.

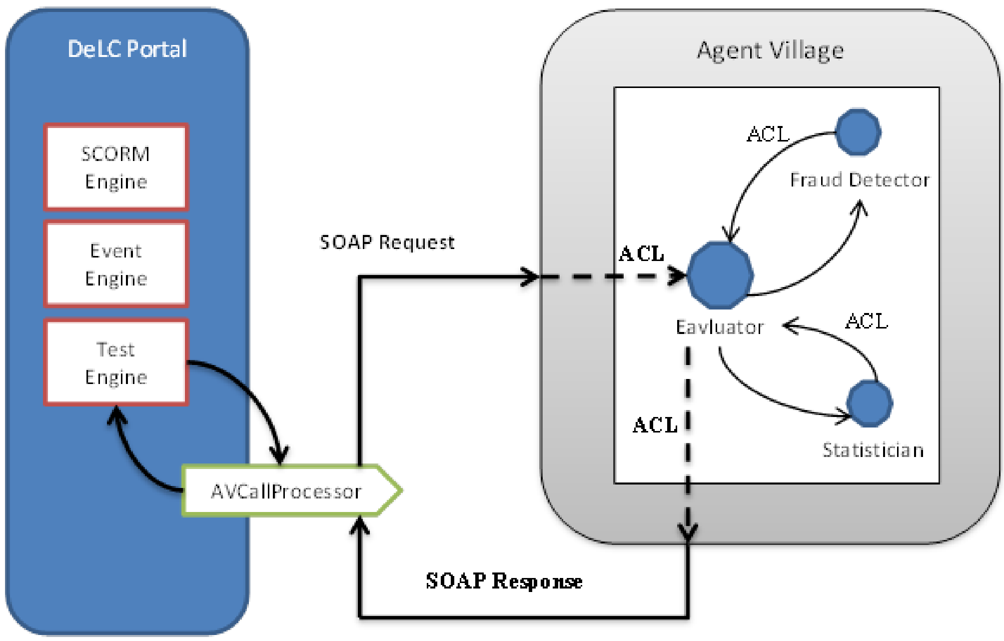

Figure 2. Architecture of the Education Cluster.

mainly for the convenient realization and integration of business functionality. Therefore, they must "transfer" the responsibility for activation and support of the connection to an active component of the architecture, as agents do. To do this, the service sends an explicit message to the agent's environment, which in turn identifies the change of environment and reacts by interpreting the message. Depending on the identified need for assistance, the agent activates the actions required. The reactive behavior of the architecture could be implemented by:

- Synchronous model – it is analogous to calling subroutines in programming languages. In this model, the service sends a message to AV and waits for the result from the corresponding agent before continuing its execution.
- Asynchronous model – in it the interaction is accomplished through a mechanism for sending and receiving messages.

The communication between the portal's services and the Agent Village's agents is achieved via web services (Fig. 2). The intelligent assistants' functions are wrapped in web service functions and thus are accessible for the portal's needs – this represents the AV's reactive behavior. On the other hand, the agent environment is active and could react to impending changes or events. Thus, the agents are able to undertake certain actions if they decide that there are proper conditions to execute them. Then, the agents send the results to the portal and, if necessary, to its users; this represents the AV's proactive behavior.

The internal communication among agents inside the AV uses ACL messages (Agent Communication Language) [12].

When a service is needed, which is functionality of some of the intelligent assistants, the portal generates SOAP Request message, which is sent to the agent's web service, whose functionality has to be exploited. This message is captured by a system component of the agent's environment, called Web Services Integration Gateway (WSIG), which is responsible for transforming the incoming SOAP Request message to an ACL message. Then this ACL message is transmitted to the proper agent, whose functionality serves the called-for web service. When the agent does its job it generates an answer in the form of an ACL message which is sent back to the environment, where a SOAP Response message is generated to be sent to the portal's calling component.

For processing SOAP messages in the portal, there is a system component, called AV-Call Processor. Its purpose is to generate a proper xml SOAP Request message, so that the right web service is called and, on the other hand, extract the result returned from the SOAP Response message and provide it to the portal in a convenient form.

In the proactive behavior (agents work "on behalf of the user"), an agent from the AV can determine that in its environment "something is happening", that would be interesting for the user, who is assisted by that agent. The agent is activated and can perform certain actions to satisfy the preferences (wishes) of the user. The agent can inform the user of its actions through the educational portal.

The difficulties associated with the management of our architecture's pro-activity, result from the fact that the portal is designed for reaction to the user's requests. Therefore, the pro-activity can be managed only asynchronously and, for this purpose, we provide development of a specialized service, which is to check a "mailbox" for incoming messages from the AV periodically.

 ISBN: 978-1-61208-185-4

IV. Assistants

A part of the functionality of the Education Cluster is to organize students' examinations – to be more precise, this is a part of the DeLC portal. It consists of automated test generation, including various questions – "choice-like", answer matching and open-ended questions. To help the teacher estimate these answers, an intelligent service is added, which automatically assesses the open-ended questions.

Another function which is also related to student examination is comparing students' answers among themselves and to Internet search engines resources. Since the portal provides an integrated chat system, the students are able to communicate through it. But during the examination it could lead to cheating. That is why an intelligent assistant is built, which tracks the chat system communications and when it detects a suspicious message, it is written in the database, where the teacher can see it at any moment. This assistant will also compare the answer to Internet resources.

To control the effectiveness of the aforementioned intelligent assistants, there is a third assistant which stores statistical data about their operation. When the collected data is sufficient, the assistant could make conclusions and give advices to other assistants.

A. *Evaluator Assistant*

The Evaluator Assistant provides expert assistance to the teacher in assessing electronic tests. A system service is built in the Test Engine for automated assessment of "choice-like" questions. In the standard version of the architecture, open-ended questions are assessed by the teacher and the mark is entered manually in the service which prepares the final assessment for the test. In the new architecture the Test Engine calls the assistant (an intelligent agent), which makes an "external" assessment of the open-ended questions.

If an "external" assessment is needed, where the Test Engine initiates a "request" for expert assistance, the reactive behavior of the EA is exploited. The EA has a wrapper (the environment of the agent) for identification, which "masks" it as a web service for the portal. The Test Engine relies on an AVCallProcessor for making the request and procession of the answer. When a request for assistance arises, this service generates a SOAP Request message and sends it to the Agent Village. When a SOAP Response is received, it parses the answer and extracts the estimated rating by the EA.

In the surrounding environment of the EA, the received SOAP Request messages are transformed into ACL (Agent Communication Language) [12] messages which are understandable for the agent. Some of the basic parameters of the messages are:

- Text, which is an answer of an open-ended\ question.
- Parameters for the estimation method used.
- Maximum number of points for this answer.

After the calculations, the EA generates an answer in the form of an ACL message, which is then transformed from the environment into a SOAP Response message (a result from a web-service call). In the answer, there is a parameter for the amount of points which is then extracted by the AV-Call Processor.

B. *FraudDetector*

The role of this assistant is using its behavior to recognize some of the most frequent kinds of fraud during examination. They include:

- Keyword guessing for a particular question;
- Copy/paste from Internet search-engines resources;
- Exchange of information among students using the portal's chat system.

This agent cooperates with the Evaluator assistant during the examination process and if it recognizes a probability for fraud, it informs the Evaluator to take action concerning a specific question, thus demonstrating its proactive behavior. All suspicious information is stored in the database for the teacher to access at any moment.

C. *Statistician*

This assistant is still in development. Its functionality will be to store information about all processes of the aforementioned two assistants: details about automated answer-assessment and about checking chat communication and suspicious messages between students.

Regarding the automated answer-assessment, this agent will need feedback about the teacher's actual assessment for each answer. Thus, it will compare the assessment of all used algorithms in the Evaluator assistant and make conclusions about their effectiveness.

V. Approbation

The education cluster is used in a real education process. Some results for the "Introduction to Databases" lecture course (IDB) are summarized in this section.

In summary, there were 142 students examined in 2 subjects: 74 studying "Informatics" (3^{rd}-year students) and 68 studying "Business Information Technologies" (2^{nd}-year students). These students answered 453 open-ended questions. 127 of the answers were blank, considered as irrelevant for the system's effectiveness test, so they were excluded from the automated-evaluation statistics, given below. These statistics concern the remaining 326 answers.

Currently, the Evaluator Assistant uses two different algorithms for answer evaluation, called word-matching and optimistic-percentage. Some results of the evaluation results are shown in Fig. 3 in comparison to the points given by the teacher.

The average number of maximum points for the answer is 4.2. That number makes sense when we calculate the number of answers, which are evaluated with a one-point difference between the agent's (automated) and the teacher's evaluation. We consider these evaluations to be very close.

The summarized results of the automated evaluation of the answers are shown in Fig. 4.

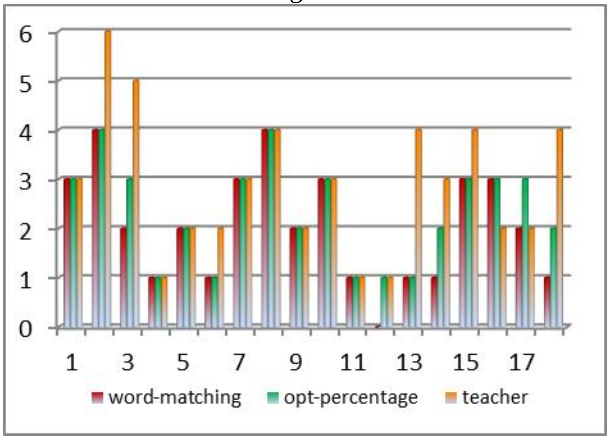

Figure 3. The points given by the two algorithms and the teacher.

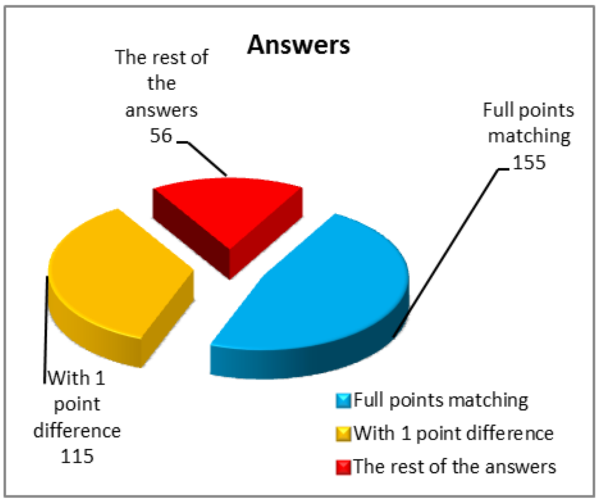

Figure 4. The results of the automated answers evaluation.

Currently, there is a new version of the evaluation algorithm. It was tested with these answers and the results were slightly better. They are shown in the Table 1 below.

TABLE I. THE RESULTS COMPARISON

Version	**old**		**new**	
Answers	*326*	*%*	*326*	*%*
Full points matching	155	47.55	162	49.70
With 1 point difference	115	35.27	116	35.58
The rest of the answers	56	17.18	48	14.72

VI. CONCLUSION

In this paper, an extension of the DeLC (Distributed eLearning Center) architecture was presented. The extension consists of making the architecture agent- and service-oriented, adding reactivity and proactivity to the existing e-learning environment. The reactive and proactive behavior of the architecture was demonstrated with the Evaluator Assistant, which provides expert assistance to the lecturer in assessment of electronic tests. Two other agents were projected for giving assistance to the educational portal's services. The entire system was tested for a year in the Faculty of Mathematics and Informatics at the University of Plovdiv "Paisii Hilendarski" (Bulgaria) in the bachelor courses of two subjects – Informatics and Business Information Technologies. The main results of the use of the cluster are increased interest and more active inclusion of the students in the education process.

ACKNOWLEDGMENT

The authors wish to acknowledge the support of the National Science Fund (Research Project Ref. No. DO02-149/2008) and the Science Fund of the University of Plovdiv "Paisii Hilendarski" (Research Project Ref. No. NI11-FMI-004).

REFERENCES

[1] Micro Soft Class Server, http://www.microsoft.com/education/products/ <retrieved: 12, 2011>.

[2] The IBM Learner Portal, https://www-04.ibm.com/jct03001c/services/learning/ites.wss/us/ <retrieved: 12, 2011> .

[3] HP Learning Center, http://h30187.www3.hp.com/ <retrieved: 12, 2011>.

[4] Moodle, http://moodle.org/ <retrieved: 01, 2012> .

[5] IMS Global Learning Consortium, http://www.imsglobal.org/ <retrieved: 01, 2012> .

[6] Advanced Distributed Learning, http://www.adlnet.org/ <retrieved: 01, 2012> .

[7] SCORM 2004, http://www.adlnet.gov/Technologies/scorm/SCORMSDocuments/ <retrieved: 01, 2012> .

[8] Stoyanov S., I. Ganchev, I. Popchev, M. O'Droma, R. Venkov, DeLC – Distributed eLearning Center, Proc. of the 1st Balkan Conference in Informatics BCI'2003, 21-23 November, Thessaloniki, Greece, 2003, pp. 327-336.

[9] Stoyanov, S., Ganchev, I., Popchev, I., O'Droma, M., From CBT to e-Learning, Journal Information Technologies and Control, No. 4/2005, Year III, pp. 2-10.

[10] S. Stoyanov, I. Ganchev, M. O'Droma, H. Zedan, D. Meere, V. Valkanova, Semantic Multi-Agent mLearning System, A. Elci, M. T. Kone, M. A. Orgun (Eds.): "Semantic Agent Systems: Foundations and Applications", Book Series: Studies in Computational Intelligence, Vol. 344, Springer Verlag, 2011, pp. 243-272.

[11] S. Stoyanov, I. Popchev, E. Doychev, D. Mitev, V. Valkanov, A. Stoyanova-Doycheva, V. Valkanova, I. Minov, DeLC Educational Portal, Cybernetics and Information Technologies (CIT), Vol.10, No 3., Bulgarian Academy of Sciences, 2010, pp. 49-69.

[12] Agent Communication Language Specifications, http://www.fipa.org/repository/aclspecs.html <retrieved: 01, 2012> .