Load Balancing Procedure for Building Distributed e-Learning Systems

Marian Cristian Mihăescu, Dumitru Dan Burdescu
Software Engineering Department
University of Craiova
Craiova, Romania
{mihaescu, burdescu}@software.ucv.ro

Mihai Mocanu, Costel Marian Ionascu
Software Engineering Department /
Analysis and Statistics Department
University of Craiova, Craiova, Romania
mocanu_mihai@software.ucv.ro, icostelm@yahoo.com

***Abstract*— Nowadays e-Learning environments manage and deliver high volumes of data. Under these circumstances it becomes highly advisable shifting towards distributed software systems. This approach may bring high benefits regarding the average response time and therefore the overall quality of the e-Learning environment. One possible solution is represented by load-balancing. This means that the workload is distributed among two or more web servers such that a faster response is obtained in proposed scalable system. The load balancing is performed according with the results obtained after putting the e-Learning system under stress that is performed by a simulation tool.**

Keywords- load balancing, e-learning, distributed systems, simulation tool

I. INTRODUCTION

This paper addresses the problem of building a high quality distributed e-Learning system. The proposed solution regards balancing the workload among two or more servers.

Currently, most of the e-Learning systems do not have a distributed infrastructure. All software components (e.g. web server, database server, objects repositories, etc.) usually reside on the same machine. One of the key issues in the design of scalable and high quality e-Learning systems is represented by the workload balancing problem. This problem consists of efficiently distributing hardware and software resources such that average system response is acceptable.

This paper presents solution for workload balancing a specific e-Learning system. For proving right decisions are taken a web based simulation architecture is used for determining the way an e-Learning system may shift towards a distributed architecture. The outcome of such a process may regard resource distribution, the quality of service offered by the system and the distributed software architecture. The simulation tool is intended to be used only in a testing environment not in a productive one. Any e-Learning platform that is scheduled for shifting towards a distributed architecture may pass through the process that is described in this paper. After the simulation process is performed there may obtained conclusions regarding the resources that need to be distributed and the fashion in which distribution needs to be accomplished.

Resource identification and distribution across different computers represents one of the main outcomes. The quality of service regards the performance, availability and reliability. The software architecture describes how the application functionality and data is distributed among available hardware and software resources.

The presented architecture enables an e-Learning system to be tested with custom designed usage scenarios. Building and running different scenarios is aimed to produce valuable information regarding how the e-Learning platform will actually perform when put under different stress conditions. The obtained information will represent the starting point regarding all decisions about how the platform will shift towards a distributed architecture.

The second section presents the state of the art regarding presented issues. The third section presents the e-Learning infrastructure that has been used for testing. Section four presents the functionality of the simulation tool. Section five present the methods that are used for building the simulation framework. Section six presents early and expected results. Finally, conclusions are presented.

II. RELATED WORK

For the some services provided by some popular web sites, the web sites may receive a huge number of client connections in a very short period. Because of this, the huge connections will incur the network congestion or rise response time and even cause the operation of web sites under the unstable situation [5]. This is why building distributed environments represents a clear objective that is spread around many different application areas. Among them, we mention civil and military distributed training applications [1], collaborative design [2], commercial multiplayer game environments [3] and e-learning [4].

The key addressed issues regard data modeling, communication model, data consistency, network traffic reduction, partitioning problems.

Data modeling presents specific modalities of distributing persistent or semipersistent data in a distributed environment. Generally speaking, data may be distributed using replicated, shared, or distributed strategy [7].

The communication model regards the way learners interact with the system in terms of performance. The employed communication model has a direct influence over response times. Different communications schemes (e.g. broadcast, peer-to-peer or unicast) determine different response times [6]. Communication model also regards establishing in details the way different software and hardware resources relate one to each other.

 ISBN: 978-1-61208-120-5

Data consistency is already defined in the area of database management [7]. Data consistency consists of ensuring that all involved parties (e.g. learners, course managers, administrators) that use the system (e.g. the e-Learning environment) must have the same vision on all the data they are working with.

Another key issue for designing a scalable and efficient e-Learning system is the partitioning problem [8]. The approach must lead to an efficient assigning of workload among resources of the system. In short, we consider that learners may be considered as clients of a distributed system. All learners compete over the same non-distributed system.

The partitioning scheme will allow the workload to be efficiently distributed among the components of the distributed architecture. The partitioning mechanism is based on determining the overall performance of the e-Learning environment taking into consideration the workload each resource in the system is assigned to but also on the inter-resource communication. A resource may be software or a hardware component.

Some methods for solving the partitioning problem have already been proposed [9], [10]. However, there are still some features in the proposed methods that can be improved. For example several heuristic search methods can be used for solving this problem [11], [12].

Blackboard e-Learning platform [13] uses load balancing within their content system in order to optimize performance and reliability [14].

A-Tutor Learning Management Tool [15] offers a hosted solution with load balancing across all systems, 24x7x365 monitoring, and nightly backups [16].

Load balancing may be found in Apache Tomcat server as well as in J2EE Web Applications. This paper presents a custom load balancing procedure for e-Learning applications in general and to Tesys [17] e-Learning platform in particular.

Designing distributed learning environments [18] is becoming increasingly popular. Many of the currently existing systems neglect the quality the offered. The offered services are loosing quality due to the increasing number of users that need to be served simultaneously and due to everyday increasing size of the data that needs to be managed.

The problem of building distributed e-Learning environments has been addressed by many important developers. This paper presents an elaborate process in the form of a load balancing procedure. The presented procedure makes sure that obtained solution is optimal for the system that is taken into consideration. This paper presents the way of obtaining a custom solution rather a general one.

The aim of the paper is to present an analysis framework which decides the most efficient way of designing the distributed system started from a non-distributed one.

III. E-Learning System

E-Learning systems are mainly concerned with delivery and management of content (e.g., courses, quizzes, exams, etc.). Since we are speaking about a web platform the client is represented by the browser, more exactly by the learner that performs the actions.

Defining the e-Learning infrastructure or the presented purpose represents the first and the most important step. In this phase, all the possible actions that may be performed by a learner need to be presented. There are also identified the resources that are delivered by the e-Learning system. Finally, there are identified the highly complex business logic components that are used when actions are performed by learners.

Each implemented action needs to have an assigned weight. In the prototyping phase, the assignment of weights is performed manually according with a specific setup. This assumes that we have an e-Learning system that is already set up. The main characteristics regard the number of learners, the number of disciplines, the number of chapters per discipline, the number of test/exam questions per chapter and the dimension of the document that is assigned to a chapter. The data that is obtained from analyzing a certain setup will represent the input data for the simulation procedure.

Another type of activities regarding learners are represented by the communication that take place among parties. Each sending or reading of a message is assigned a computed average weight.

A sample e-Learning setup infrastructure may consist of 500 students, 5 disciplines, 5 to 10 chapters per discipline, 10 to 20 test/exam questions.

For this infrastructure here may be established a list of costs for all needed actions that may be performed by learners. The weight assigned to an action takes into consideration the complexity of the action and the dimension of the data that is obtained as response after the query is sent.

For obtaining reasonable weight, a pre-assessment procedure is performed. The simulation tool performs this procedure from a computer that resides in the same network as the server such that response times are minimal. Each request that is composed and issued to the e-Learning platform is measured in terms of time and space complexity. A scaling factor will assign each action a certain weight such that the scenarios that will be created when real time testing starts will have a sound basis.

The pre-assessment procedure firstly loads all the data regarding the analyzed e-Learning platform. This means the data about all managed resources (e.g. disciplines, chapters, quizzes, etc.) are loaded such that the simulation tool may build valid requests for the e-Learning environment.

IV. Simulation Tool

The simulation tool emulates one or more learners that access the e-Learning platform in a specified timeframe. The setup of the tool defines the stress level that put on the e-Learning platform.

The main parameters that set up the simulation tool are: *timeframe*, *noOfLearners*, *stressLevel* and *weightLevel*. Timeframe represents the period for which the simulation tool will issue requests. *noOfLearners* represent the number of learners that issue requests for the e-Learning platform. *stressLevel* represents the level of stress put on the platform by learners. The stress level may be from 0 to 3 where 0 represents a low stress level and 3 represents a high stress level. When stress level is set to 0 this means that each learner will issue one request per minute. When stress level is set to 3 this means each learner will issue 7 requests per minute. *weightLevel* represents the level of weight put on the platform by learners.

The weight level may be from 0 to 3 where 0 represents a low weight level and 3 represents a high weight level. The weight level represents the average weight for actions that are performed in a certain amount of time. According with the weights presented in table 1, the level 0 weight level corresponds to 5 units per minute and level 3 weight level corresponds to 500 units per minute.

The main goal of the simulation tool is to perform the actions of one or more learners. Instead of having the platform tested by real students the simulation tool is used. There are many advantages of this approach although there are not used real data. The main advantage regards flexibility and scalability. The behavior of the simulation tool is managed through setup parameters. Thus, the level of stress that is put over the system may be easily set and thus the behavior of the e-Learning environment under certain conditions may be evaluated. The first outcome of the simulation tool is in the form of a custom usage scenario that complies the setup parameters. This scenario consists a suite to http requests that are to be issued by a certain number of clients.. The simulation tool is regarded as a multithreaded client where each thread will represent a learner that interacts with the e-Learning platform. The main features and software components are: pre-assessment procedure and the following software components: virtual learner, scenario builder and dispatcher.

The pre-assessment is the first performed. Its main task is to build an in-memory representation of the assets that are managed by the e-Learning environment. The outcome of this step produces an XML file with all assets (e.g. disciplines, chapters, quizzes, etc.). The data representing existing assets will be used by the scenario builder module to create valid HTTP requests. Pre-assessment step also computes average weights for HTTP requests. This step along with loading the data about managed assets and the configuration file represent the setup phase.

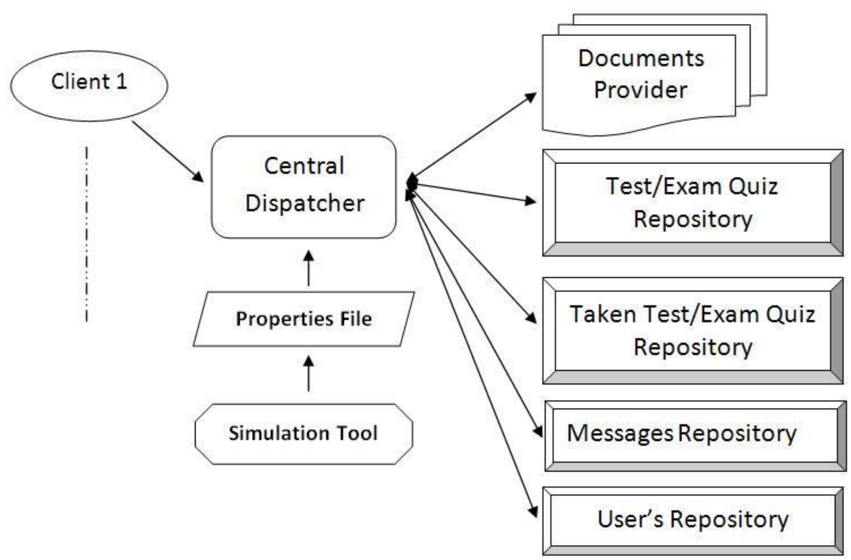

Figure 1. The architecture of a distributed e-Learning system

Building the XML file with assets representation and the setup is performed as a scheduled job with a predefined frequency. This approach is due to the fact that the assets are dynamically created by the simulation tool itself. For example, in a scenario may be included that HTTP requests that are needed for simulating a testing or an examination. One of the outcomes of this procedure is a record in the database with the data regarding the taken test. Unfortunately, this test may not be yet included in scenarios the want to display tat taken test. At the next running of the pre-assessment procedure the data about the taken test will be included in the XML file that contains the assets and thus the action that represents the display of that test may be included in one or more scenarios.

A virtual learner software component is regarded as an autonomous agent. In fact, the virtual learner represents a complex entity that performs predefined custom business logic within the environment. The set of all actions that will be performed constitute its behavior. Modeling the behavior is the task of the scenario builder component. The learner itself is regarded as a software component that interacts with the e-Learning environment. The main functionalities implemented for learner software component regard issuing HTTP requests, getting the HTTP response and logging the transfer data. Request and response date and the amount of transferred data are among the data that are logged.

The scenario builder software component is responsible for building the scenarios that will run against the tested e-Learning environment. The input for this module is represented by the XML file containing the assets, the file containing the weights for each action that may be performed and the configuration file. The configuration file contains the values for the parameters that are to be used when creating the scenarios. The main parameters that may be found are presented in section 4 as general setup parameters of the simulation tool.

The dispatcher software component is responsible for managing and coordinating learners. The input data is represented by the scenario that is build by the scenario builder component. The dispatcher creates the pool of learners that will issue requests and receive responses from the e-Learning environment. It will also gather and centralize the data from all learner components that interact with the e-Learning environment in a structured format.

V. Load Balancing Architecture

Here will be presented the architecture of the distributed e-Learning system. The distribution is performed by using several web servers. The servers have full duplicated data content. This means each server has the ability to answer to any of the possible requests. The decision to have a full duplicated data content rises a very difficult problem regarding data synchronization. For this prototype solution we use an active replication scheme in which clients communicate by multicast with all replicas. The full duplicated data scheme enables the system to answer the client from any replica. Still, the goal of the obtained load balancing architecture is to allow a shift towards a passive replication scheme. This means certain type of requests will always end up being answered by a certain replica that manages the needed data. In this process, the simulation tool has an important role. Each issued request may generate a signature that contains an indication regarding the replica and the asset that has been accessed. The signatures may thereafter be used for assessing the quality of the distribution process and therefore the quality of the load balancing architecture.

Figure 1 presents the architecture of the entire system where different web servers and the relationships between them are presented. There are considered five different types of requests. Each web server has the ability to answer to any request that may be sent to the central dispatcher. The dispatcher works as a scheduler between clients and web servers. The architecture virtually consists of 5 web servers that work as: (1) documents provider server, (2) test/exam quiz repository, (3) taken test/exam repository, (4) messages repository and (5) users repository. All 5 servers work as a web server cluster and perform their activities in a transparent manner for the users

When the clients issue requests, the requests are delivered to the central dispatcher first and then forwarded to the backend web servers for processing. Thus, it is important for us to design an efficient load balance algorithm in the dispatcher in order to direct the incoming requests to backend web servers. Synchronization issues are also managed by the central dispatcher.

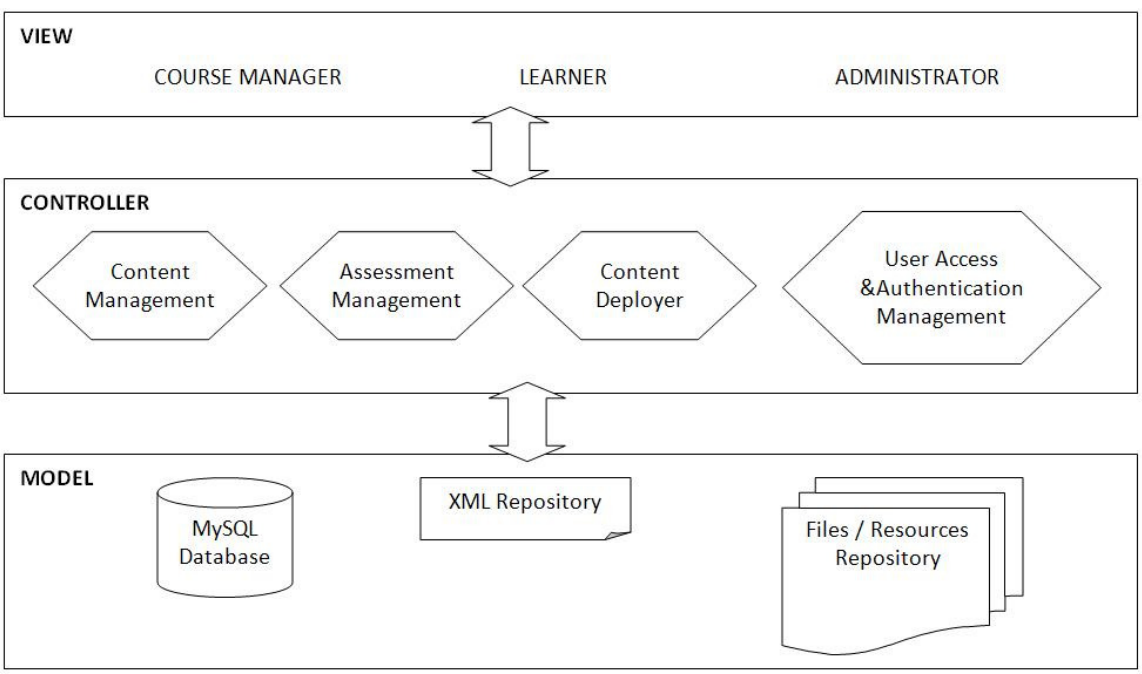

Figure 2. The MVC architecture of the e-Learning system

Under above presented circumstances it is obvious the importance of load balancing algorithm. The algorithm will decide which resource is to be allocated for performing the requested task. According with presented architecture, the algorithm is highly dependable upon the content/service that is needed. The requests are differentiated based on the contents needed by clients.

The documents associated to courses are regarded as static content. Whenever a static content is requested, its associated cost is estimated. The cost value is added in the weight matrix to the position representing the assigned web server. Once the job is finished the cost value is subtracted from the matrix. The matrix that holds the weights is called load balancing matrix. In presented procedure, pos represent the position within load balancing matrix which holds the minimal value. If *load-balancing-matrix[pos]* has value 0 this means the web server associated with position pos is 100% available for incoming task. If *load-balancing-matrix[pos]* is greater than 0 this means the web server associated with position pos is handling a request of weight equivalent with the stored value. The web server has itself a maximum available computational power expressed as a weight that may be found in a properties file. The properties file holds the maximum computational power values for each web server that will be part of the load balancing infrastructure. If the available weight is greater that the weight of the incoming task than the task may be assigned to the corresponding web server.

The procedure dispatchTask, which runs over load balancing matrix, is:

```
procedure dispatchTask (task t){
    cost = computeCost(t);
    pos = findMinimumValue(load-balancing-matrix[]);
    if (cost + load-balancing-matrix[pos] > MAX_ALLOWED_COST)
        throw("WAIT");
    else {
        load-balancing-matrix[pos] += cost;
        StartTask(t, WebServer[pos]);
    }
}
```

Figure 3. The procedure for dispatching tasks

The incoming requests are classified according with the five category set presented in Figure 1. The requests that need to access and download documents or messages are classified as static requests. The requests that need to access and manage test/exam quizzes or users are classified as dynamic.

The metrics that define the weight of an action that is performed on a web server relate to the average number of requests of a certain type in a specified timeframe. Another metric is represented by the number of requests of a certain type that are executed at a certain moment by a server. This represents the currently executed tasks on each server. Another metric is defined by the estimated weight of a certain task. The weight value is pre-evaluated during simulation.

Figure 2 presents the architecture of the e-Learning platform from the MVC (Model-View-Controller) point of view. Within each layer there are presented the main building blocks in a non-distributed fashion. This view of the e-Learning platform is the starting point in the distribution process. The simulation tool must take into consideration the number of modules from each level and the needed distribution granularity.

For the implementation of e-learning system a distributed architecture is proposed. In this design, multiple E-Learning modules provide e-learning functionalities to clients. In this architecture, we have designed our e-learning system with 3-layer architecture: View Layer, Controller Layer and Model Layer. The view and controller layers are deployed on an application server. Model layer represents the persistent storage for data and is represented by a database, an XML repository and a file/resource repository.

The view layer is represented by the user interface of system and provides the interface of the tools and applications of the e-Learning platform.

In controller layer, the logical and control functions of e-learning system are implemented. In addition, this layer provides access for data source (mySQL database, XML repository and files/resources repository) for view layer and connection between view layer and model layer.

The Content Service Requester is implemented in E-Learning Service Provider and initiates interaction with Content Service Provider. This layer uses the web service to provide data sources for higher layer.

Obtaining content from data sources (data bases, repositories of XMLs or files) is implemented within the control layer in the form of a data access layer. The communication between layers is accomplished in a client/server fashion. The data sources are the persistent and heterogeneous data repository, which usually are designed with the focus on distribution.

VI. Conclusions and Future Work

In this paper we presented an e-Learning infrastructure for which a load balancing infrastructure is designed. For designing a feasible architecture a simulation tool is presented. The tool acts as a pool of virtual users. The actions are performed according with a specified properties file. There were identified 5 type of actions: file transfers, test/exam quiz management, taken test/exam quiz management, messages management and users management. That is why the proposed architecture consists of 5 web servers that contain full duplicate data and business logic. In front of them there was placed a central dispatcher that actually performs the load balancing procedure.

The central dispatcher uses as input data a properties file that is filled with data by the simulation tool and by the system administrator. The business logic is mainly concerned with finding the appropriate web server that will execute the needed action.

The adoption of distributed design regards the future possibility of integrating additional functionalities without affecting the overall quality of service offered by the platform. This design meets the scalability requirements. Integrating a service within the overall design (e.g. a

 ISBN: 978-1-61208-120-5

recommender service) may be accomplished with minimal affecting of the overall response times.

As future work we plan: (i) provide a scaling procedure such that the central dispatcher may work with any number of web servers; (ii) enhance the simulation tool such that weight task estimations to be more accurate; (iii) to extend the design for integrating other services (e.g. a recommender service) within the architecture; (iv) change the active replication scheme to a passive replication scheme in which clients communicate with a distinguished replica; (v) have an evaluation strategy for the system that has been shifted towards a distributed architecture. We shall report on our progress in subsequent papers.

ACKNOWLEDGMENT

This work was supported by the strategic grant POSDRU/89/1.5/S/61968, Project ID61968 (2009), co-financed by the European Social Fund within the Sectorial Operational Program Human Resources Development 2007 – 2013.

REFERENCES

[1] D.C. Miller and J.A. Thorpe, "SIMNET: The Advent of Simulator Networking", Proc. IEEE, vol. 83, no. 8, pp. 1114-1123, 1995.

[2] J.M. Salles Dias, R. Galli, A.C. Almeida, C.A.C. Belo, J.M. Rebordao, "mWorld: A Multiuser 3D Virtual Environment", IEEE Computer Graphics and Applications, vol. 17, no. 2, pp. 55-65, 1997.

[3] M. Lewis and J. Jacboson, "Game Engines in Scientific Research", Communications of the ACM, vol. 45, no. 1, pp. 27-31, 2002.

[4] C. Bouras, D. Fotakis, and A. Philopoulos, "A Distributed Virtual Learning Centre in Cyberspace", Proc. Fourth Int'l Conf. Virtual Systems and Multimedia, VSMM '98, pp. 18-28, 1998.

[5] M. Zari, H. Saiedian, and M. Naeem, "Understanding and Reducing Web Delays", IEEE Computer, Vol. 34, No. 12, pp. 30-37, 2001.

[6] Y. Kawahara, H. Morikawa, and T. Aoyama, "A Peer-to-Peer Message Exchange Scheme for Large Scale Networked Virtual Environments", Proceedings of the The 8th International Conference on Communication Systems (ICCS'02), pp. 957-961, 2002.

[7] Philip A. Bernstein, Vassos Hadzilacos, and Nathan Goodman, Concurrency, Control and Recovery in Database Systems. Addison-Wesley, 1997.

[8] C.S. John, M.F. Chan, K.Y. Oldfield, K.Y. So, and T.S. Tam, "Dynamic Partitioning for a Distributed Virtual Environment", Department of Computer Science & Engineering, Chinese University of Hong Kong , 1998.

[9] Jonh C.S. Lui and M.F. Chan, "An Efficient Partitioning Algorithm for Distributed Virtual Environment Systems", in IEEE Transactions on Parallel and Distributed Systems, Vol. 13, No. 1, pp. 193-211, 2002.

[10] Sandeep Singhal and Michael Zyda, Networked virtual environments: design and implementation, ACM Press/Addison-Wesley Publishing Co. New York, USA. 1999.

[11] P. Morillo and M. Fernandez, "A GRASP-Based Algorithm for Solving DVE Partitioning Problem", Proceedings of the 17th International Symposium on Parallel and Distributed Processing, pp. 22-26, 2003.

[12] P. Morillo, M. Fernandez, and J.M. Orduna, "A Comparison Study of Moder Heuristics for Solving the Partitioning Problem in Distributed Virtual Environment Systems", Proceedings of the 2003 international conference on Computational science and its applications, Springer-Verlag, pp. 458-467, 2004.

[13] Blackboard Inc., http://blackboard.com [retrieved: December 8, 2010].

[14] Lara Oerter and Deborah Everhart "Blackboard Content System™ - Product Overview White Paper", Blackboard Inc., http://library.blackboard.com/docs/contentsystem/Content_System_wp.pdf [retrieved: December 8, 2010].

[15] ATutor is an Open Source Web-based Learning Content Management System (LCMS),http://atutor.ca/ [retrieved: December 8, 2010].

[16] EduTools Course Management System Comparisons, Product details, http://www.edutools.info/compare.jsp?pj=4&i=624 [retrieved: December 8, 2010].

[17] D.D. Burdescu and M.C. Mihăescu, "Tesys: e-Learning Application Built on a Web Platform", Proceedings of International Joint Conference on e-Business and Telecommunications, Setubal, Portugal , pp. 315-318, 2006.

[18] M. Ganzha, "E-learning with Intelligent Agents," IEEE Distributed Systems Online, vol. 9, no. 2, art. no. 0802-o2004, 2008.

.

ISBN: 978-1-61208-120-5