Hypermedia System for Online e-Learning and e-Testing in Project Management

Eugen Zaharescu[1], Georgeta-Atena Zaharescu[2]

(1) "OVIDIUS" University of Constanta
124, Mamaia Blv., Constanta 900527, ROMANIA
E-mail: ezaharescu@univ-ovidius.ro
(2) "DECEBAL" High School, Constanta

Abstract

The main purpose of this paper is to present a flexible hypermedia system for online e-learning and e-testing in the domain of project management education. This virtual learning environment uses the newest open source web technologies available for the moment: PHP, MySQL, AJAX, XML and XSLT. The paper describes the implementation principles of this e-education system structured on two functional sections: hypermedia-based e-learning and online dynamically adapted e-testing. In the first section, a powerful database system allows online access to a very well organized video tutorial, covering the main aspects of project management e-learning. In the second section, tests are automatically generated and adapted for each student as they are presented as shuffled sequence of questions and alternative answers, using a web-based interface.

***Keywords**:* e-Learning, e-Testing, Distance Learning, Hypermedia, Computer-Assisted Education, Educational Platform, Virtual Learning Environment.

1. Introduction

The hypermedia system for project management education, presented in this paper belongs to the wide domain of *e-learning* and *e-testing* (or *e-assessment*). The most important feature of this modern education domain is the emergence of the **ICT** (*Information and Communication Technologies*) use in knowledge production, diffusion, consultation and automatic assessment. The generalization of the use of **ICT** in e-learning, leads to an explosion of **LU** (*Learning Units*) on the internet. Indeed, many studies [3; 6; 7] reveal hundreds of **LMS** (*Learning Management System*) able to provide **LU** for e-learning, but these are not always reusable by other **LMS**.

In the last decade, two approaches have tried to answer to this problem of **LU** reuse. The first approach is to create repositories of **LU** shared on internet like research projects ARIADNE, COLIS, Edusource, DLESE and MERLOT. The second approach is to reuse the educational scenario as a whole.

Also, new educational languages, standards and specifications, like **IMS** (*Instructional Management Systems*), **EML** (*Educational Modelling Language*) and **MISA** (*Méthode d'Ingénierie des Systèmes d'Apprentissage*), propose models for educational scenarios design and reuse.

The emergence of the **ICT** leads to an explosion of the web based tools and services (forum, chat, **LMS**, etc.) which are not always interoperable. The new web technologies and **W3C** recommendations represent a solution for the interoperability of these tools and services and the development of a virtual exporting/importing space.

The architecture of our hypermedia system for project management education is structured in two sections:

1. The first section is a **LMS** (*Learning Management System*), based on a set of web services. The hypermedia catalogue developed in this section is able to share the **HLU** (*Hypermedia Learning Units*) located in local or distant hypermedia repositories.
2. The second section is an **AMS** (*Assessment Management System)* based on a dynamic tests generator and a statistical evaluator.

This proposed e-learning and e-testing (e-assessment) management system can perform automatic test generation and grading from a wide variety of question types supported. It is a data driven system which can dynamically adjust e-test contents and parameters and fast reorder the whole e-test form by random number generation.

The structure of this paper is as follows: section 2 presents the concepts and principles upon learning management system is based on; section 3 describes the functional architecture of learning management system; section 4 shortly presents the implemented learning unit catalogue and repository; section 5 presents the concepts and principles of this web based e-testing system; section 6 describes Web based e-testing system architecture; in the end the experimental and statistical results followed by final conclusions are revealed in section 7 and 8.

2. Learning Management System. Concepts and Principles

The modelling language, proposed by the **IMS** (*Instructional Management Systems*) Global Learning Consortium is widely inspired from **R. Koper** works [2]. It provides a rich terminology which allows to describe in a formal way and to implement reusable educational scenarios. Also, it offers an educational flexibility because the designer can describe every type of **LU** (*Learning Unit*) (e.g. lessons, problem based learning, etc.).

An **LU** is introduced by **IMS** (*Instructional Management Systems*) Global Learning Consortium as an abstract item which makes reference to an element of learning or education as for example a lesson or a module [1]. It is to note that an **LU** represents more than an orderly collection of resources; it also includes a variety of prescribed activities (e.g. search activities, evaluation activities, training activities, etc.), the services, the tools and the resources produced by the learners and the staff.

The activities, the roles, the resources and the workflow depend of the ones from the others in the educational scenario.

Conceptually, an **LU** is modelled as a content package containing the educational scenario. The content of **LU** is built according to the **IMS** content package. It is composed of the following two major components [1]:

1. The first important component of **LU** is the **manifest** which describes the content structure and the associated resources. It is an **XML** (*eXtensible Markup Language*) file, called *"imsmanifest"*. The element **<manifest>** is the root of the manifest file. It contains three direct children elements:

1.1. The first child is an optional element, called **metadata**. It describes the manifest as a whole and uses the IEEE-LOM [4] metadata scheme.

1.2. The second child is called **organizations**. It describes how the content is organized to be delivered to the learners. To create the educational scenario for the LU, the **<organizations>** element includes the **<learningdesign>** element. This last one contains the elements which describe the educational scenario. It summarizes the idea according to which the educational scenario takes place as a theatre play. The educational scenario is organized in acts in which the activities are proposed to the roles in a computer environment consisted of learning objects and of services (chat, forum, e-mail, etc.). It is designed to allow reaching the learning objectives. It is described according to the hypothesis of some prerequisites which a learner must have to realize the activity. The educational scenario is organized in A, B and C three levels [1]. The level A is constituted by the general description elements of the educational scenario. While the B level, adds to the A level, the elements of the educational scenario personalization (conditions and properties). Finally, the level C, adds the notification mechanism which allows making dynamic the educational scenario.

1.3. The last child is called **resources**. It is a collection of references to resources. The element **<resources>** consists of several (zero or more) **<resource>** element. A resource is not necessarily composed by a single file. It can be also constituted by a set of files. Each file of **<resource>** element is represented by **<file>** element. These files can be internal files referenced by relative address or external files referenced by **URL** (*Uniform Resource Locators*).

2. The second important component of **LU** is represented by the physical files (local or external files) associated to the very contents of **LU,** itself. They are electronic representations of media, such as text, sound, images, animations, graphs or any piece of data that can be rendered from the Internet and presented to learning subjects.

Each of these media may have multiple digital formats (e.g. **WAV**, **MP3** or **WMA** for sound files, **.AVI, .WMV, .MPG** or **.SWF** for video files). A physical file can be created by the **LU** designer or reused from a repository.

The internal files must be included inside the **PIF** (Package Interchange File) file. The **manifest** file and all other **XML** control files (**DTD, XSD**) identified by the manifest must be placed at the root of the **PIF** file, which is a concise web delivery zip format. The use of a concise zip format facilitates and accelerates the transport of the PIF file over the Internet.

3. Functional Architecture of Learning Management System

The most important architectural elements of Learning Management System are:

- the **actors** representing the persons who play the various administrative, educational, technical roles.
- the **LMS** representing the learning management and the organization sub-system or core system. It will realise management of the learners, individualization of the learning, evaluation of the learners, etc.
- the **CMS** (*Content Management System*) representing the **LU** management system. It helps to create, to updates and to manage the **LU**.

These two sub-systems are based on two principles. The first principle is the separation of the contents and the form. It allows the designers to concentrate on the design and the creation of contents without worrying by the form. Some **CMS** proposes predefined models which the designer uses to insert their contents. The contents consist from the existing resources (reuse) or created from the new resources. The second principle is the import and the export of the **LU**. A **LCMS** offers both **LMS** and **CMS** combined functionalities **(LCMS = LMS + CMS)**.

- The **LU repositories** are data bases containing **LU**. They also implement web services which allow their interoperability with the catalogue, the **LMS** and the **CMS**.
- The **catalogue** is the tool which allows sharing of the **LU** on the network. Also, it allows searching the **LU** on **LU repositories** according to some search criteria. The **LU** which answers to search criteria is downloaded from the **LU** repositories. They are then used by the **CMS** or by the **LMS**.

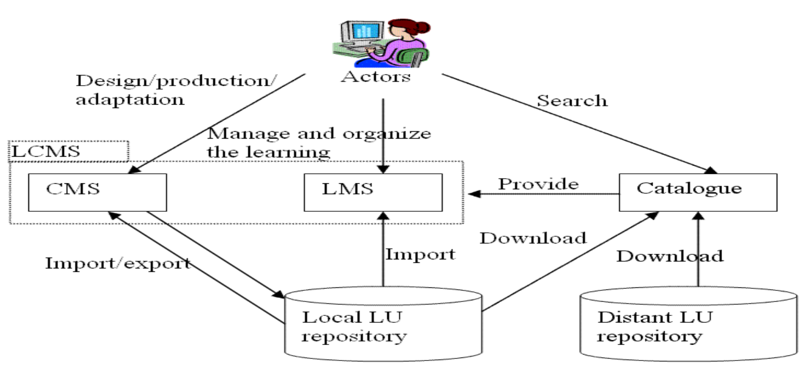

Figure 1. The functional architecture of **LMS** (*Learning Management System*)

4. Learning Unit Catalogue and Repository

There are two types of **LU** repositories: *local* and *distant*. The local repository is a data base which is on the system server and contains the **LU** created by actors. They can be imported or modified by **LMS** or **CMS.** The distant **LU** repository is a data base containing **LU** located on another web server. The **LU** belonging to this data base is downloadable by the **LU** catalogue module, only.

Figure 2. Learning unit catalogue and repository during training process

5. Web Based e-Testing System. Concepts and Principles

Definition 1

Question bank is the core of every e-testing system and is represented by a database of unique questions with parameters, from which the test generation module will make simple selections.

The necessary question parameters may be:

- *question type* according to possible answer/answers;
- *question weight/value* for final summarized grading;
- *question domain/area/section* following the theory classification etc.

The architecture of e-testing systems should have the following modules:

- *question input module* with special forms for question entry process;
- *question importing module* from other similar e-testing systems database;
- *question removing module* from own database.

In order to create a an exchanging space for importing/exporting questions and to provide compatibility between different systems, several standard structuring forms have been developed for the elements from question banks. Among them, the most important and promising standard is developed by **IMS** (*Instructional Management Systems*) **Global Learning Consortium, Inc.** http://www.imsglobal.org [1]. These system standard specifications are defined in XML, following W3C Consortium recommendations. The **IMS** Question & Test Interoperability (**QTI**) specification describes a data model for the representation of questions and test data and their corresponding results reports. Since 2005 starting with version 2.0, **QTI** supports parameterized questions via assessment item templates [1].

The e-testing system question bank may have two different types of questions:

- *fixed answer question (objective question)* and
- *free answer question (unobjective question)*.

The *fixed answer question (objective question)* is made up of a text body for problem description and a list of possible answers, where the student must choose from, the correct one/ones. Most of the e-testing systems use this type of questions in assessment process.

The taxonomy of *fixed answer questions* may be subsequently defined like this:

- *multiple answer question* is the most used type in automated assessment.
- *short answer question* requires an short text answer to be provided;
- *short text* or *numerical value answer question* requires a computed result;
- *hot-spot question* or *visual/interactive answer question* requires an object/position identification, graphical element connections, etc.

Furthermore, for *multiple answer question* category we may have several variations:

- **Yes/No** *or* **True/False** *answer question* has only two opposite alternative answers;
- **MC/SA***(Multiple-Choice/Single-Answer) question* has only one correct answer;
- **MC/MA***(Multiple-Choice/Multiple-Answer) question* has more correct answers;
- **Priority Setting/Selection** *answer question* require items ranking.

The *free answer questions* (*unobjective questions*) have no predefined answer. They are usually used when assessing higher levels of learning domains or Bloom's taxonomy: **Cognitive(Knowledge)-**mental skills, **Affective(Attitude)-**growth in feelings or emotional areas and **Psychomotor(Skills)-**manual or physical skills. The *free answer question (unobjective question)* category may be divided in two sub-types/sub-categories: *program code answer question* and *essay answer question*.

In the end, it may be concluded that using question banks instead of static test creation method, we obtain significant advantages in e-testing system:

- a wide set of possible learning objects to choose from in assessing process;
- test generation time is decreased very much.

Definition 2

Test creation algorithm represents the questions selection process from the question bank (system core), followed by the generation of student's presentation form. There are 5 different test delivery models [8], depending on the characteristics of the tested subject knowledge: linear, dynamic linear, testlets, mastery models, adaptative:

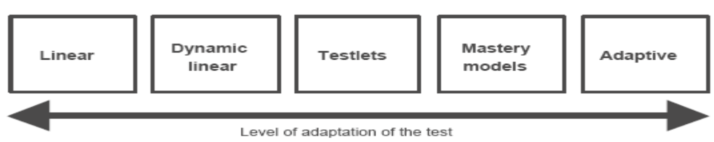

Figure 3. Test delivery models

Linear tests are not adaptive to users and consist of predefined questions and predefined order. Assessment can be done automatically and results be summarized.

Adaptive tests depend on student's knowledge. The parameters of test generation are defined dynamically during the test according to given answers.

Definition 3

Grading and results reporting are the final actions of the automatic e-testing system which will display the results of the assessment immediately after the end of assessment process, when all answers of all questions are definitely entered.

The evaluation of the entered answers can be made in two different moments of time: *at the end of the entire test* if the system let the subject the possibility to change the entered answers or *after each answer* is entered otherwise.

6. Web Based e-Testing System Architecture

The design approach of **TestManager** system is based upon a three layer architecture: *database layer* (it stores the *question bank* on a database server and communicates with web server to generate dynamic web pages), *application layer* (it receives requests from user interface layer and generates appropriate answer to every user request; it is executed on web server) and *user interface layer* (used by users to submit requests through application logic layer; it is executed on every client station).

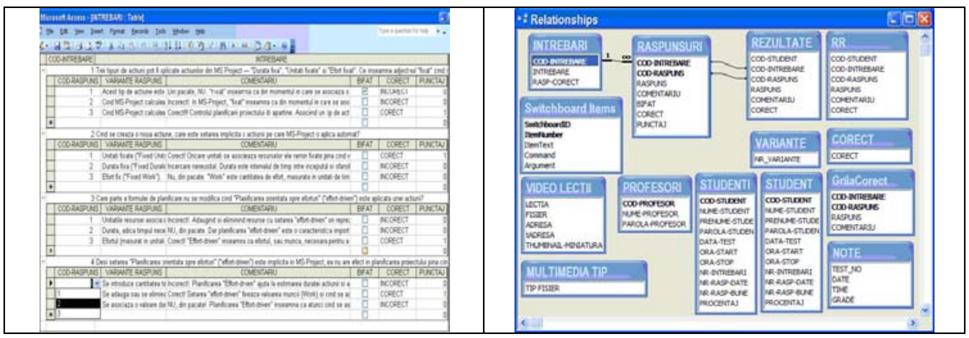

Figure 4. E-testing system question bank and database tables structure (*database layer*)

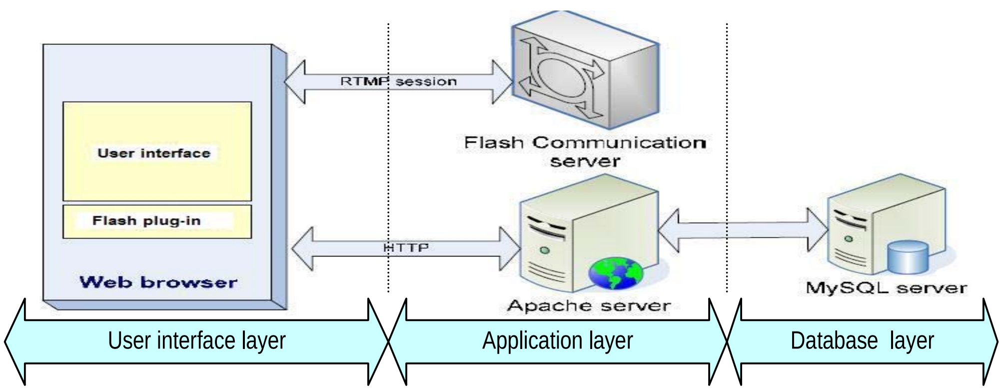

Figure 5. The three layer architecture of **TestManager** system

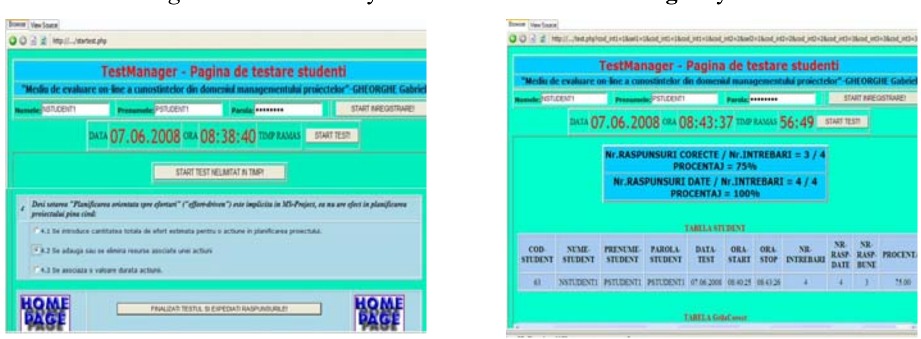

Figure 6. Assessment process and final statistical results reports (*user interface layer*)

7. Experimental and Statistical Results

The e-testing system implements automatic evaluation at the end of the test, when the knowledge tested subject finalize himself the answer entry process. The system displays subject final results compared with the complete set of correct answers.

We have implemented *negative grading,* in order to eliminate intelligent or lucky guessing. So, *the number of points given for any true answer selection* (m_T) and the *number of points taken for any false answer selection* (m_F) will be given by the expressions:

[1] $$m_T = \frac{m}{n_T} \quad , \qquad m_F = -\frac{m}{(n_A - n_T)}$$

where n_A = total number of possible answers, n_T = total number of true answers and m = maximum grade or total number of test points. The final calculated grade will be:

[2] $$M(N_T) = N_T * m_T + (n_T - N_T) * m_F = N_T * \frac{m}{n_T} + (N_T - n_T) * \frac{m}{(n_A - n_T)}$$

where N_T = total number of true answer selections.

The final grade of every test will be displayed as either, absolute value and relative value (percentage of possible points).

We have repeatedly evaluated a sample of 10 subjects with randomly generated tests using both normal and negative grading. Number of incorrectly passed test by intelligent or lucky guessing, instead of real knowledge, tends to lower with the increasing number of questions per test and with the negative grading, as shown in the table 1:

Table 1

Number of incorrectly passed test by intelligent or lucky guessing

Grading Type	**Number of questions / test**		
	40 questions	*50 questions*	*60 questions*
Normal Grading	9 %	6.7 %	4.5 %
Negative Grading	4 %	1.7 %	0.9 %

8. Conclusions

This paper presents and analyses the main features of **TestManager**, a hypermedia system for e-learning and e-testing dedicated to project managers. The design of **TestManager** conforms to the models and specifications provided by **IMS** (*Instructional Management Systems*) Global Learning Consortium.

Therefore, the system can be easily adapted to any education domain because it produces reusable **LU** (*Learning Units*). The Web based e-assessment section of **TestManager** has a data driven system design as the main concept behind this system is the question bank. In fact, this represents the system core, where questions are selected from, during test generation process. The test creation algorithm dynamically generates equally weighted tests according to previously predefined strategy. Finally, the *negative grading* method statistically eliminates intelligent or lucky guessing answers.

REFERENCES

[1] IMS Content Packaging Information Model, Learning Design Information Model: http://www.imsglobal.org/content/packaging http://www.imsglobal.org/learningdesign

[2] KOPER, R. (2005), *Modelling Units of Study from a Pedagogical Perspective, the pedagogical metamodel behind EML*, http://eml.ou.nl/introduction/docs/ped-metadodel.pdf.

[3] LMS Open Source, https://fnl.ch/LOBs/LOs_Public/OpenSourcePlatf.htm

[4] LOM metadata official web page, http://ltsc.ieee.org/wg12/files/LOM_Final.pdf

[5] MERRILL, M.-D. (1994), *Principles of Instructional Design*, New Jersey Educational Technology Publications.

[6] Oravep Study, http://www.educnet.education.fr/superieur/plateforme.htm

[7] Thot official web page: http://thot.cursus.edu/

[8] PATELIS, T. (2006), *An overview of computer-based testing. The college board*: http://www.collegeboard.com/research/html/rn09.pdf

[9] PETTIGREW, M. (2001), *Random guessing on multiple choice tests*, http://www.shu.ac.uk/services/lti/people/mp/mcq/

[10] Web Page of LORNET project, http://www.lornet.org/index.htm