

LEARNING DESIGN MODULE IN SCORM E-LEARNING ENVIRONMENT

Elena Shoikova, Malinka Ivanova

Department of Electronics, Technical University of Sofia, “Kliment Ohridski” 8, 1000 Sofia,
Bulgaria, Tel.: + (359 2) 962 40 49; 965 21 40,
e-mail: shoikova@tu-sofia.bg, m_ivanova@tu-sofia.bg

The present paper describes a project started at the Technical University of Sofia Research & Development Laboratory “E-Learning Technologies” aimed at a Learning Design Module (LDM) development as a part of authoring tool in open source based Shareable Content Object Reference Model (SCORM) – compliant e-learning environment. The design methodology comprises an object-oriented approach, Unified Development Process (UDP), Use Case Analysis, Unified Modeling Language (UML) as well as a client-server architecture solution. Applying the following technologies does the implementation of the LDM: object-oriented PHP language, HTML, JavaScript, HTML, XML and relation Database (DB).

Keywords: Learning Design Module, Development, IMS LD specification

1. INTRODUCTION

To be successful, e-learning must offer effective and attractive courses and programmes to learners, while at the same time providing a pleasant and effective work environment for authors who have the task of developing course materials, planning the learning processes, providing tutoring, and assessing performance. The systematic process of translating general principles of learning and instruction into plans for instructional materials is called LD [1]. Also that the term is used to refer to the product of the design process, i.e. a LD defined, in the IMS LD Specification as “a description of a method enabling learners to attain certain learning objectives by performing certain learning activities in a certain order in the context of a certain learning environment” [2]. Rob Koper’s article on pedagogical meta-modeling describes LD as “modeling units of study” [3]. The concept of Unit of Learning (UOL) is central and the notation of UOL is called Educational Modeling Language (EML). A UOL is an abstract term used to refer to any delimited piece of education or training, such as a course, a module, a lesson, etc. A UOL represents more than just a collection of ordered resources to learn - it includes a variety of prescribed activities, assessments, services and support facilities provided by staff members.

LD provides a conceptual model for the description of teaching and learning processes based on EML [4]. With LD it is possible to develop and present advanced, interoperable e-learning courses. The present paper describes a project started at the Technical University of Sofia R&D Laboratory “E-Learning Technologies” and aimed developing a LDM as a part of a SCORM-compliant e-learning Environment. LDM is implemented using object-oriented PHP language, HTML and JavaScript.

The module is web-based with client/server architecture and as part of e-learning environment works with MySQL Database.

2. NEEDS ANALYSIS AND REQUIREMENT DEFINITIONS

Using LD, authors are able to talk in terms of pedagogy rather than technology, helping to create high quality UOLs. The made analysis of LD Tools points out the importance of working with exact tool in the process of creation the learning scenarios. This analysis summarizes the best practice in LD Tools and gives possibility to defining the requirements to LDM in the situation of engineering education. Spectrum of tools may be established, going from those which are presented in terms and structures which remain close to the IMS LD Specification, and those which are presented in non-formal colloquial terms. Some popular at this moment LD Tools used by authors of UOLs at design time are grouped as Tree-based Editors: RELOAD [6], aLFanet LD Editor [7], Komposer and High level Editors: MOT, LAMS [8]. Tree-based Editors present the elements of LD as a branching tree. An interface is provided to enable the instructor to navigate through a tree structure that directly reflects the IMS LD Specification, adding parameters and resources. To this extent they are close to the Specification, but they are some distance away from the base line. For some purposes tree-based editors will not be satisfactory, much support is provided for the user, and an interface will be required which is further from the Specification. In this case are used High level Editors.

The analysis of these LD Editors exposes some drawbacks:

- In the Low level Editors the attention of users is focused on detailed recognition of LD Specification and XML language instead of learning process.
- Extended functionality of High level Editors that provide users through while LD process that require more time and attention to learn, setup, control and navigate the tools instead of creating concrete UOLs.
- End-user download, installation and maintenance procedures that waste time require system administrator's attention.
- The adaptation of these LD Editors for specific learning processes and exact pedagogic experience in more times is impossible.
- There are the difficulties in point view of integration of LD Editor in entire e-learning environment.

Now days the main strategy in designing the support applications is focused on these that hide complexity from the user with interoperability between different stages in learning design process. Therefore the LDM has to be designed as a High Level Specific Purpose Tool, and Distance from IMS LD Specification.

The LDM requirement specification defined at different level of details in the context of needs is the following:

- Functional: template-based LD Editor with user-friendly GUI, that guides authors in designing process of UOLs, with constrained pedagogy, distant from Specification, allows the sequencing of the activities and binding with resources, points to services;

- Operational: web-based LD Editor with client/server architecture, with end user installation and maintenance free, easy integration in e-learning environment, standard protocols and technologies used, based on EML and LD Specification, SCORM e-learning content used;
- Management/organizational: services of e-learning environment are used.
- Technological: for easy integration in existing e-learning environment LDM has to develop in object-oriented PHP language with JavaScript, with MySQL Database communications, output is LD XML file, http/ftp protocols used.
- Standards based: used LD IMS Specification and EML, used SCORM e-learning content.

3. DESIGN METHODOLOGY

The design methodology comprises: pedagogy concepts from EML, UDP for producing the working software, Use Case Analysis that elaborate requirements to software system, notation UML for visualizing the behavior of the system, an object-oriented approach in realization of programming code, IMS LD specification for binding designed UOL with XML, as well as a client-server architecture solution. The latest software design and development methodologies are fundamentally changed in order to minimize the risk of failure and to deliver successful solutions faster [9]. The UDP is used as the main development strategy, relies on the incremental process model. In the project, UDP is chosen because: it is planned and managed; it is predictable; it accommodates changes to requirements with less disruption; it is based on evolving executable prototypes; it is risk driven. The UDP consists of 4 phases: Inception, Elaboration, Construction, and Transition. During Inception, the requirements of the whole project is reviewed, focus is concentrate on discovering risks and proposing a suitable software architecture that will support all the functional and non-functional requirements. Elaboration is the phase where start the first development iteration. In this phase is developed the highest risk requirement and is attempted to prove the software architecture. Construction is where all remaining functionality is built. At the end of this phase, there is a completed beta-version of the software. Transition is where the software is installed and undergoes end-user acceptance testing. At the end of this phase, the project is complete.

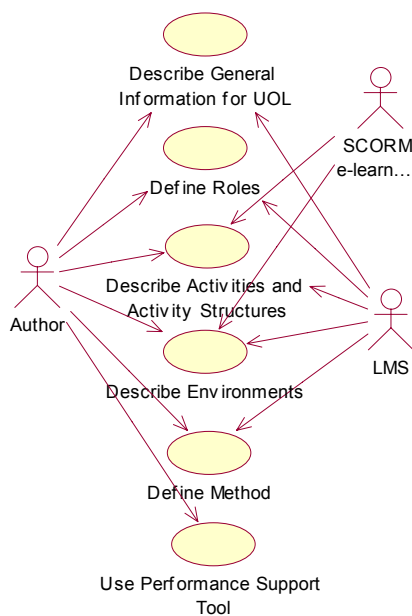
The main stages, which are iterated through each increment, are: iteration planning, requirements capturing, analysis&design, implementation, test and prepare release. The UDP is a sum of various activities, needed to satisfy the requirements of the system users. It is built around use cases as the primary functional component.

3. USE CASE ANALYSIS

The Use Case analysis is the most efficient method for requirements capturing and software system functional specification. It is used to define the features of software application. The Use Case analysis also helps to layout the actors or users and their role in running the system.

The main actors that interact with the LDM are as follows: Author-human actor and non-human actors - Learning Management System (LMS) and SCORM e-learning Content Editor.

Conceptual strategy for building UOL is depicted in Figure 1 by six Use Cases: *Describe General Information for UOL*, *Define Roles*, *Describe Activities and Activity Structures*, *Describe Environments*, *Define Method* and *Use Performance Support Tool* that are examined in Table 1. In the functional model of LDM Author interacts in all of six Use Cases. The Use Case *General Information for UOL* is used to form instructional content and metadata about UOL. The descriptive element that metadata contains may be used for searching in digital repository. Prerequisites express the necessary requirements for starting a UOL, or an activity within a UOL. One or more Learning Objectives can be defined for each UOL and for each activity within the UOL.



The *Roles* are used to specify the actors that play different roles in a UOL. To invoke a learning process one or more activities are assigned to the certain Role. Different sequences can be defined within *Activity Structures*. The *Learning Environments* specify which objects and services are needed for the entire set of activities. *Method* is used to describe how activities to be performed in a particular environment by different Roles. The Author uses *Performance Support Tool* to help and guide himself in time of working with LDM. SCORM e-learning Content Editor is powerful tool in process of SCORM content creation. LMS gives and supports LDM with services.

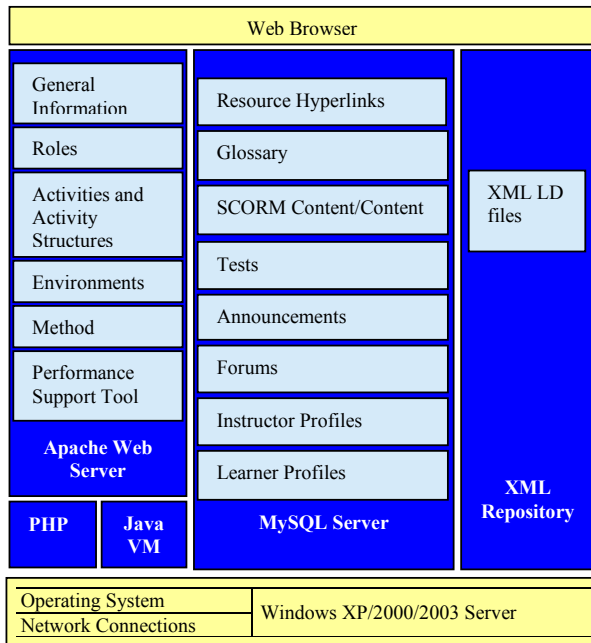
Figure 1 Use Case Model of LDM

Use Case	Description
Describe General Information for UOL	In this Use Case the elements Title and Summary of the General Information are used as instructional content for describing the UOL. All elements – Title, Summary, Keywords, and Creator define the metadata of UOL. The Author describes Learning and Prerequisites. LMS as actor uses this information to provide different services to users.
Define Roles	The Author specifies the actors that play a role in a UOL, as uses registered in LMS users and describes their roles – learner, teacher, expert, tutor, and instructor.
Describe Activities and Activity Structures	The Author describes for each Activity - Title, what should be done learner, how should it be done and define when an activity is to be considered completed. The Author assigns to the activity before this defined learning objective(s), prerequisites and resources. SCORM e-learning Content Editor is used for the creation of e-learning resources. LMS manages activities and activity structures.
Describe Environment	The Author describes learning environments in which the activity should take place. The learning environment consists of tools and services in order to support the learner and/or staff. The objects including in the learning environment are: Knowledge Object, Announcement Object, Tool Object, Communication Object, and Test Object. They are managed by LMS.
Define Method	The Method is important for interpreting the UOL by LMS and in this way it becomes interactive object. It includes the set of activities or activities structure to be performed (in a particular environment), as well

	as a way of representing these activities and objects in the environment to learners or staff members. The Author defines Activity Structures. Within these structures activities can be grouped and put in a fixed sequence. The Author assigns activities to specified roles.
Use Performance Support Tool	An Electronic Performance Support Tool is the primary delivery mechanism that provides assistance to the Author through the some components. The first component in gaining proficiency is training. The second component is support with information, expert advices and instructions. The third component of proficiency is experience.

Table 1 Use Cases

4. FUNCTIONAL ARCHITECTURE



The developed functional architecture of LDM is presented in Figure 2 and describes its components: software component which is managed by Apache Web server, Database managed by MySQL Server and XML Repository with XML LD files. The LDM is integrated part of SCORM-compliant e-learning environment that includes LMS, SCORM-compliant Learning Content Management System (LCMS), SCORM E-learning Content Editor, Digital Repositories (MySQL DB, SCORM content, XML LD files), and Run Time Environment (RTE).

Figure 2 LDM Functional Architecture

5. IMPLEMENTATION OF LDM

Implementation is concerned with building, testing, and deploying the new LD software application. Specifically, this activity includes detailed design of each LDM component (user interface and programmes), construction, and integration of the code and user interface of LDM in SCORM-compliant e-learning environment. Implementation is largely dependent on the architectural choices middleware and on used web e-learning environment. For realization of LDM has been chosen object-oriented PHP language, HTML and JavaScript in order it to be modular part of existing open source-based SCORM-compliant e-learning environment. It is developed as a distributed application. In Figure 3 is proposed the technical architecture of LDM as a client/server approach with three bases flows. The software module is placed on server side. The Author works only with web browser and populates the Form' fields in order to design/edit UOL on client side. The Author temporary saves new data in DB tables, in order to uses them in different Form' fields in the designing process of UOL. For this purpose have been designed some new DB tables which to temporary store the objective, the specific objectives, prerequisites, etc. The first flow reflects on data movements between web browser and MySQL DB. Some LD components as activities and environments need to be

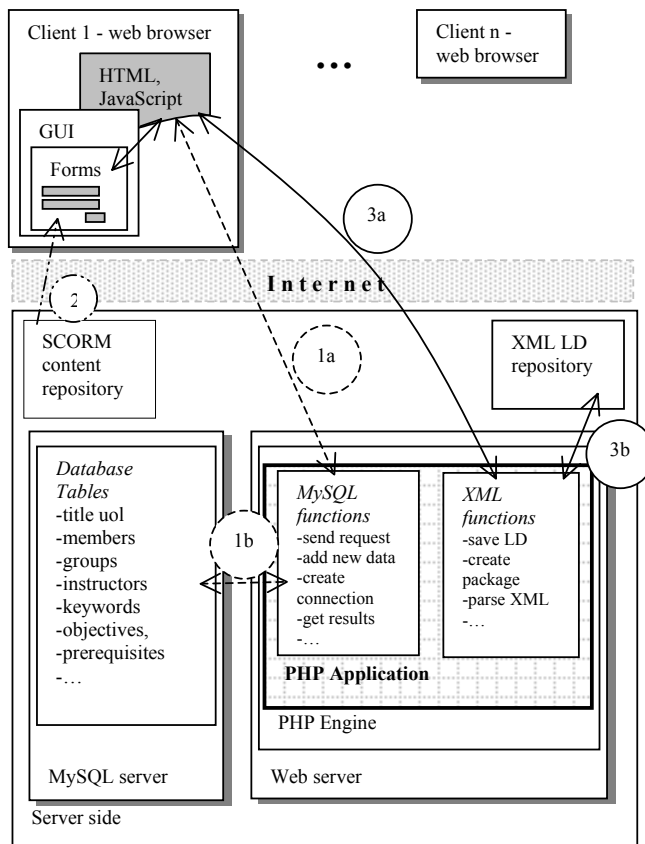


Figure 3 Technical Architecture

and software system functional specification is defined through Use Case Analysis as well as detail examination of functions. The functional architecture of LDM is proposed that describes its software components. For integrating the module in existing e-learning environment is proposed Architecture of SCORM-compliant E-learning Environment. The developed Technical Architecture of LDM presents the realization of this project as client/server approach. GUI is constructed and software application is coded in PHP language, HTML and JavaScript.

7. REFERENCES

- [1] CETIS, *Learning Design inspiration*, 2003, <http://www.cetis.ac.uk/content2/20031105152011>
- [2] IMS Global Learning Consortium, Inc., *IMS Learning Design Information Model, XML Binding Document, Best Practice Guide*, 2003
- [3] Koper, R., *Modeling units of study from a pedagogical perspective: The pedagogical meta-model*, 2001
- [4] Wilson S., *Europe focuses on EML*, CETIS staff, 2002
- [5] Advanced Distributed Learning, *SCORM Overview*, 1.3.2, July 22, 2004
- [6] RELOAD Project, 2004, <http://www.reload.ac.uk>
- [7] Carrion J., C. Fuentes, *General System Architecture and Design. AL-Fanet*, SAGE, Madrid, 2004, <http://hdi.handle.net/1820/258>
- [8] Koper R., C. Tattersall, *Learning Design*, Springer, 2005
- [9] Kratchen F., *Rational Unified Process*, Addison-Wesley, 2002

bundled with resources and the second flow presents the inserting of SCORM content in LD. The movement from the design of UOL to the XML document instance according to IMS LD Specification is shown by flow three.

6. CONCLUSIONS

A significant step has been taken for implementing new LDM as a part of SCORM-compliant e-learning environment for the purposes of higher engineering education and training in Technical University of Sofia. In paper is made analyses of existing LD Editors and is summarized the best practice in LD Tools that gives possibility to defining the requirements to LDM. The main picture of LDM' features