

## **eJCSE: SCORM-COMPLIANT eLEARNING MODULE IN “SOFTWARE ENGINEERING”**

**S. Stojanov<sup>1</sup>, E. Doychev<sup>1</sup>, M. Trendafilova<sup>1</sup>, I. Ganchev<sup>2</sup>**

<sup>1</sup>E-Commerce Laboratory, Faculty of Mathematics and Informatics,  
University of Plovdiv, 236 Bulgaria blvd., 4000 Plovdiv, BULGARIA

<sup>2</sup>Department of Electronic and Computer Engineering,  
University of Limerick, National Technological Park, Limerick, IRELAND  
{stani,emil.doychev,malina.trendafilova}@ecl.pu.acad.bg; ivan.ganchev@ul.ie

**Abstract:** This paper presents a possible approach for transformation of an existing JCSE eLearning module into SCORM-compliant one (i.e. eJCSE) required for the integration into an eLearning infrastructure called Distributed eLearning Center, which aims to provide a distance learning facility available to all registered users at any place and at any time. Two types of supporting tools (an authoring system and a run-time environment, RTE) are described. Finally a review of the current state of the DeLC system and its supporting tools is made.

**Keywords:** eLearning, DeLC, SCORM, reengineering, software architectures, Java, J2EE, JetSpeed framework

### **1 Introduction**

An eLearning module in “Software Engineering” has been developed as part of an international JCSE project involving the following partners: Humboldt University of Berlin, University of Novi Sad, University of Skopje, University of Belgrad, University of Plovdiv et al [1]. The current version of this module (JCSE) includes 30 topics and over 2000 slides in .ppt format. The second version (eJCSE) has to be compliant to the SCORM standard in order to be integrated into an eLearning infrastructure called Distributed eLearning Center (DeLC).

The Shareable Content Object Reference Model (SCORM) [2] represents a pedagogically neutral model for designers and implementers of instructions to aggregate learning objects for the purpose of delivering a desired learning experience. SCORM is considered to be an important first step toward liberating learning contents from local implementations, which will free learning objects from specific learning context thus allowing them to be easily shared across multiple eLearning systems, e.g. aggregated in DeLC.

The DeLC aims to provide a distance learning facility (through a wide range of electronic learning solutions such as Web-based courseware, discussion groups, live virtual classes, video and audio, Web chat, simulations and mentoring), which is availa-

ble to all registered users at any place and at any time [3, 4, 5]. The initial goal of this center is to deliver modules, courses and degrees to individuals and groups of students who interact with educators and with organized learning materials, both in real-time (synchronous mode) and delayed-time (asynchronous mode). DeLC will initially supplement the modules taught as part of the various courses at the University of Limerick (Ireland) and University of Plovdiv (Bulgaria).

In this paper we present a possible approach for transformation of an existing SCORM-incompliant eLearning module JCSE into SCORM-compliant one (called eJCSE), which then is integrated into DeLC. This transformation concerns the following two development aspects (and appropriate tools):

- creation of SCORM-compliant learning contents;
- interpretation of SCORM-based learning structures and execution of specified sequencing behaviours.

The former problem affects the usage of the authoring systems. According to SCORM specification the learning contents and the learning activities may be created manually or by using authoring systems. We have chosen the second approach and are going to design a complete SCORM-based authoring environment. The latter problem addresses the development of the run-time environment (RTE), which can interpret and navigate SCORM structures. The two types of supporting tools (i.e. an authoring system and a RTE) will be developed and tested primarily for the JCSE.

## 2 From JCSE to eJCSE

The primary aim of the JCSE project was to create a joint lecture module in “Software Engineering” delivered in a classical style. In order to test and validate the first version of the DeLC architecture we decided to use the existing lecture contents (in .ppt format). The eJCSE is a SCORM-compliant version of the JCSE, which will be accomplished as an electronic service (eService) and integrated into the DeLC architecture. To achieve this we propose an approach comprising the following steps:

- *Reengineering* - the basic learning objects of a lecture (part of the existing SCORM incompliant eLearning module) are extracted and described in suitable object formats;
- *Mapping*– the extracted objects and their type descriptions (meta-objects) are mapped into SCORM structures;
- *Transformation*– the objects’ content is transformed into SCORM-compliant content;
- *Integration*– the new SCORM-compliant version of a lecture is integrated into the DeLC architecture.

Currently we are investigating appropriate tools supporting this approach, e.g.:

- *Object Extractor* – needed for the *reengineering* and *mapping* phases (with full automatic support if possible);

- *SCORM Lecture Builder* - needed for the *transformation* and *integration* phases. The builder is developed as a part of the DeLC authoring system.

Moreover the run-time system of the DeLC should also be modified in order to be able to accept the SCORM structures. A common approach for full adaptation of the DeLC architecture to the SCORM standard is given in [6].

### 3 DeLC authoring system

The DeLC authoring system addresses mainly the eEducator services (e.g. the preparation of electronic lecture module etc.) according to the *content/nature* classification of the DeLC eServices. There are four main classes of eServices in this category:

- *eLearning Services* - intended mostly for students but could be used also by educators. These eServices are used mainly in on-line mode processing in response to students requests.
- *eEducator Services* – intended for educators preparing necessary information resources for the eLearning Services, adding new eServices, modifying or removing existing eServices, interacting with class, groups or individual students, etc. Many of these eServices are used in off-line mode.
- *Administration Services* – a set of eServices, which are intended for the administrators in order to maintain the system itself.
- *System Services* – the primary function of these is to organize access and support of other eService types. The system services are directly integrated into the DeLC node architecture.

Our first vision for the architecture of the DeLC authoring system is shown in Figure 1. The authoring system consists of two main parts: Visual SCORM Lecture Builder and Lecture Library.

The Visual SCORM Lecture Builder is a set of visual tools that support the educators during the preparation of the electronic versions of the lecture modules. The builder provides different eEducator services, e.g.:

- *Creating* a new lecture – according to SCORM standard a lecture can be created from different sources defined in various formats;
- *Deleting* an existing lecture;
- *Modifying* an existing lecture;
- *Building* a new lecture from two or more existing lectures;
- *Grouping* a set of lectures into a lecture module – different module architectures are possible (i.e. linear, tree, network etc).

The outcomes of the builder are SCORM-compliant structures, which can be interpreted by the DeLC run-time system. The generated SCORM units are kept in an appropriate data store called Lecture Library.

#### 4 DeLC run-time environment

The SCORM run-time module (RTM) is the counterpart of the SCORM builder with a basic function to interpret the SCORM-compliant structures stored in the Lecture Library. The RTM includes an Activity Tree Builder (ATB) and a Sequencing Engine (SE) (Figure 2). Depending on the required eService, the ATB generates a plan (in the form of an activity tree) needed for the successful completion of the request. The plan is then executed by the SE.

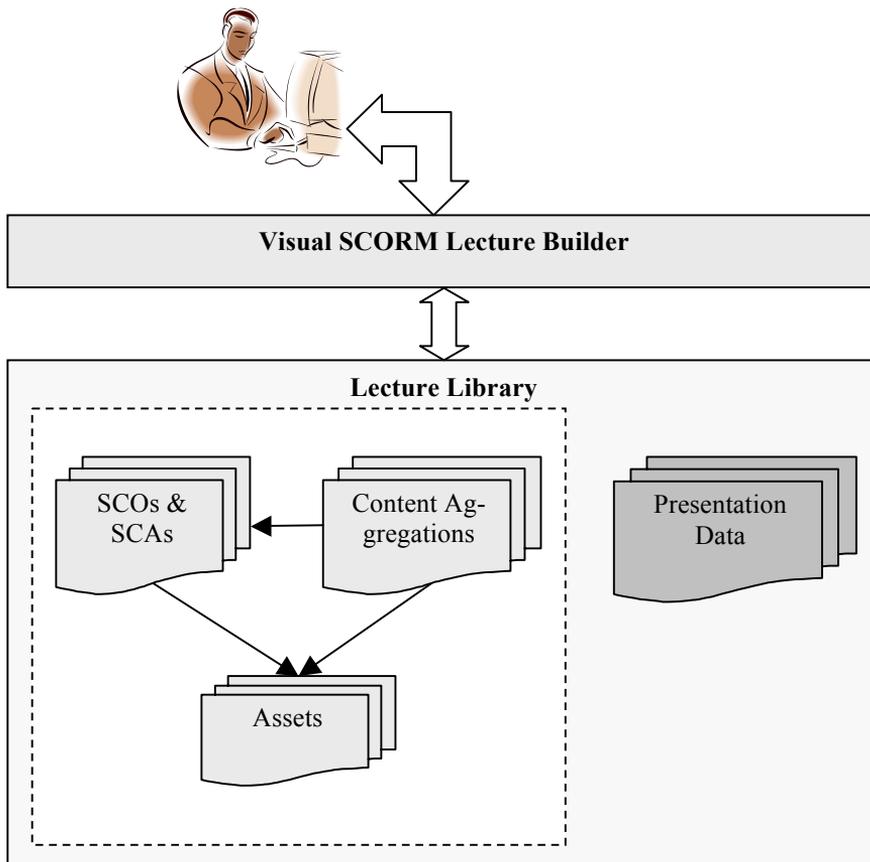


Fig. 1: DeLC authoring system

#### 5 Conclusion

Developing a Distributed eLearning Centers (DeLCs) that will offer in integrated way electronic services (eServices) available from geographically spread eLearning systems is a very complex, sophisticated and time-consuming process even though using the advantages of the IT developing tools. The first version of our center (J2EE-based [7, 8, 9]) is currently tested. The development of the second (SCORM-compliant)

version of DeLC is in progress. The SCORM-compliant DeLC run-time module will be integrated into the intelligent portal of the system implemented by using the JetSpeed framework [10]. The authoring system will be developed and integrated as an independent part of the DeLC development environment.

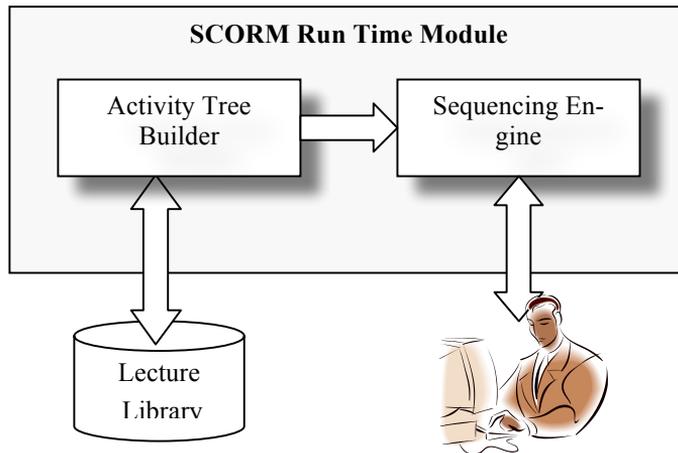


Fig. 2: SCORM Run-Time Module

## 6 References

1. Bothe, K., Schuetzler, K., Budimac, Z., Zdravkova, K., Bojic, D., Stojanov, S.: Technical and Managerial Principles of a Distributed Cooperative Development of a Multi-Lingual Educational Course, 1<sup>st</sup> Balkan Conference in Informatics (BCI'2003), Thessaloniki, Greece, 21-23 November 2003.
2. Advance Distributed Learning Initiative (ADL), Sharable Content Object Reference Model version 1.2, <http://www.adlnet.org/> (to date).
3. O'Droma, M., Ganchev, I., Stojanov, S., McDonnell, F.: "UML Design Approach to a Virtual University Information System". In Proceedings of 13th Annual EAEEIE Conference, April 2002. York, UK (2002) B-7:1-9.
4. Stojanov, S., O'Droma, M., Ganchev, I.: "A Model for Integration of Electronic Services into a Distributed eLearning Center". In Proceedings of 14th EAEEIE International Conference, Gdansk, Poland. (2003) A17:1-7.
5. Stojanov, S., Ganchev, I., Popchev, I., O'Droma, M., Venkov, R.: DeLC – Distributed eLearning Center, 1<sup>st</sup> Balkan Conference in Informatics (BCI'2003), Thessaloniki, Greece, 21-23 November 2003.
6. S. Stojanov, I. Ganchev, M. O'Droma, An adaptation of the DeLC architecture to the 'SCORM' Standard, Fourth International Conference on Informatics and Information Technology (CIIT'2003), Bitola, Macedonia, 11-14 December (in the same issue).

7. <http://java.sun.com/j2ee/docs.html> (to date).
8. Java™ 2 Platform, Standard Edition, v1.4 API Specification (J2SE specification), <http://java.sun.com/j2se/1.4/docs/api/index.html> (to date).
9. Java™ 2 Platform, Enterprise Edition Technical Overview (J2EE Overview), <http://java.sun.com/j2ee/white.html> (to date).
10. <http://jakarta.apache.org/jetspeed/> (to date).