# AN ADAPTATION OF THE DeLC ARCHITECTURE TO THE 'SCORM' STANDARD

**S. Stojanov[1], I. Ganchev[2], M. O'Droma[2]**

[1]E-Commerce Laboratory, Faculty of Mathematics and Informatics,
University of Plovdiv, 236 Bulgaria blvd, 4000 Plovdiv, BULGARIA
[2]Department of Electronic and Computer Engineering,
University of Limerick, National Technological Park, Limerick, IRELAND
stani@ecl.pu.acad.bg; {ivan.ganchev,mairtin.odroma}@ul.ie

**Abstract:** A services-oriented education system, Distributed eLearning Center - DeLC, is presented in this paper. An essential disadvantage of the current version of the DeLC is that the system doesn't support some of the existing eLearning standards. The emerging standard SCORM has been chosen for the implementation of the next version of the center. We present our first vision for adaptation of the DeLC architecture to the SCORM standard as regards the implementation of eLearning services, the model used for overall control and management, and the process of eServices integration. As necessary action we consider also changing the structure of the meta-level and the profiles-supporting system.

**Keywords:** eLearning, eServices, DeLC architecture, SCORM

## 1 Introduction

The Distributed eLearning Center (DeLC) presented in this paper may be viewed as an example of a Network-Based Education [1], where interaction between cooperative physically dispersed programs, tools, students, educators, and administrators takes place. A services-oriented approach was chosen for the development of the DeLC as an open infrastructure for offering electronic services (eServices) deployed on different servers with an inbuilt capacity for partial automatic-controlled integration (within the frames of predefined virtual structures). The DeLC is a collaborated project established between the Department of Electronic and Computer Engineering at the University of Limerick and the eCommerce Laboratory of the Faculty of Mathematics and Informatics at the University of Plovdiv. An essential disadvantage of the current version of the DeLC is that the system doesn't support some of the existing eLearning standards, e.g. the Shareable Content Object Reference Model – SCORM, [2].

The emerging standard SCORM is considered to be an important first step toward liberating learning contents from local implementations. This will free learning objects from specific learning context allowing they to be easily shared across multiple eLearning systems. SCORM is a collection of so called “books” presenting a “library” of technical specifications and best practices for implementing a SCORM compliant eLearning systems and building SCORM compliant learning content. We have chosen this standard for the realization of a SCORM-based version of the DeLC, where we hope to gain some important features, e.g.:

- *Reusability* – learning contents can be used in various applications and other educators can adapt these contents to their courses;
- *Access* – independent from place, time, and source;
- *Search and navigation* – with more precise query results and possibilities to support students’ self-study process;
- *Interoperability* – different platforms can cooperate with other systems and authoring tools.

In this paper we present our first vision for adaptation of the DeLC architecture to the SCORM standard as regards the implementation of eLearning services, the model used for overall control and management, and the process of eServices integration. As necessary action we consider also changing the structure of the meta-level and the system supporting different types of profiles.

## 2 DeLC

The Distributed eLearning Center aims to provide a distance learning facility available to all registered users at any place and at any time. This center is being established with an initial goal of delivering 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). The DeLC proposed here is described in more details in [3, 4, 5, 6].

The development of the DeLC is decomposed into the following four subprojects:

- Realization of the DeLC infrastructure;
- Modelling and implementing a set of eServices;
- Implementation of a standard DeLC Node architecture;
- Cluster building support.

The DeLC infrastructure can be represented as a graph. The arcs of the graph define relations between different nodes called DeLC Nodes (DeLCNs), which could for instance consist of a University entity’s eLearning facilities. Entities could be laboratories, departments, faculties, colleges, or the whole university itself. The DeLCNs would provide apart from a complete learning cycle a set of eServices in flat or hierarchical structure. Different DeLCNs provide different eServices, e.g. (i) a university entity provides the registered students with a completely structured programme (across modules and semesters including lectures, tutorials, laboratories, midterm tests, final examinations etc) much of which will be provided directly through the DeLC; (ii) a laboratory entity provides the registered students with a cycle of laboratory experiments. The DeLCNs are uniquely identified/marked so that users elsewhere may be made aware of their specific features. Different virtual network (or subnetwork) structures may be defined over the graph by taking into account the possibilities to access, or even exchange, eServices provided by different nodes. All DeLCNs have uniform standardized architecture. DeLC supports an integration of eServices limited within the framework of the specified eLearning clusters. This integration is realized on the basis of a model, which uses Web Services [7].

One of the main goals of the DeLC is to specify a framework for the realization and integration of different types of eServices. This framework consists of two main parts:

- eServices classification (*meta-model*) which builds a functional framework to enable an eLearning business model development;
- A set of models of the supported eServices (*subject-models*).

Both parts are created by using the Unified Modelling Language (UML) [8, 9].

As regards the *meta-model* the eServices are classified according to their content/nature, type of invocation, and mobility features. Each eService exists within a 3D discrete space defined by these three characteristics (Figure 1).

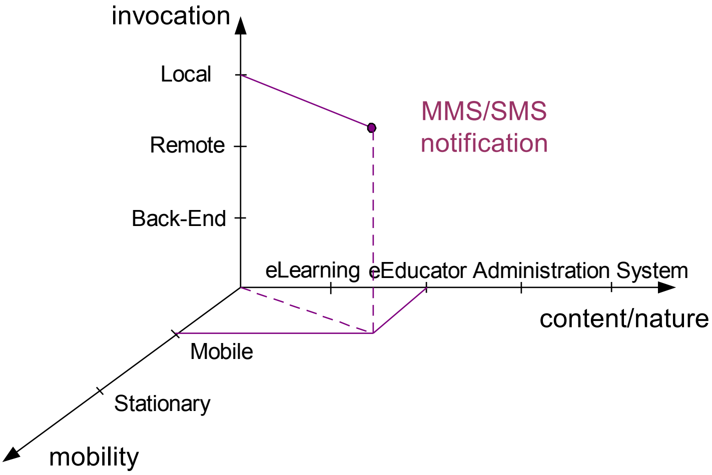


Fig. 1: eServices' 3D discrete space

According to the *content/nature* the eServices are grouped into the following four classes:

- *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 to maintain the system itself.
- *System Services* – with primary function to organize access and support of other eService types.

The eServices can be classified by *type of invocation* as followed:

- *Local eServices* – located on the local DeLCN and accessed directly from the node.
- *Remote eServices* – located on the remote DeLCNs and accessed through a Remote DeLCN Gateway.
- *Back-End eServices* – typically provided by the back-end applications (faculty administration systems, libraries etc), which are integrated into the DeLC. The access to them is provided through an appropriate Back-End Gateway.

According to *mobility features* the eServices are divided in two main groups:

- *Mobile eServices* – for users (with mobile devices) engaged in time-critical and goal-driven tasks. This requires development of a new subgroup of *context-aware* eServices that understand the users' context (e.g. location, courses/modules, classmates, educators etc).
- *Stationary eServices* – for ordinary users without a need of using mobile devises.

## 3 SCORM Specification

The current SCORM version is 1.2 [2], comprised of three core documents/books:

- SCORM overview - presents the vision for development of worldwide eLearning industry and active market of context independent eLearning objects created and stored – ready for use and reuse in different eLearning environments, achieving different learning objectives. The rationale for SCORM and a summary of the other two core SCORM documents form the rest of this document.
- SCORM Content Aggregation Model - defines the requirements for building, describing (by using meta-data) and gathering together learning objects into cohesive learning units (e.g. modules, courses, and curricula). Doing so will allow unobstructed usage of SCORM compliant learning contents, developed by using different authoring tools, on every SCORM compliant Learning Management System (LMS). The SCORM Content Aggregation Model is based on four standards and specifications:
    - *The IEEE Standard for Learning Object Meta-data (LOM)*, [10] - defines a meta-data dictionary that can be used to describe a learning object so that it can be stored into a repository, exposing its meta-data for discovery by eLearning object search engine, therefore enabling its sharing and reuse;
    - *The IMS Learning Resource Meta-data eXtensible Mark-up Language (XML) Binding Specification* [11] - says how an item from the SCORM

  information model should be presented into XML by keeping the same relations between items as they were previously defined into the model;
  - *The IMS Content Packaging specification* [12] – specifies how to collect learning content into packages that can be imported, exported, aggregated, and disaggregated by any SCORM compliant LMS;
  - *The Content Structure derived from AICC* - a graph with nodes representing the learning resources and arcs representing all possible routes that an LMS can navigate through at a run-time.
- SCORM Run-Time Environment – entirely based on the functionality defined in AICC CMI Guidelines for Interoperability [13]. The main role of the run-time environment is to provide an implementation-independent protocol for interaction between learning resources and LMS needed in order to be able to share and re-use learning objects supplied by different vendors and developed by using different authoring tools. The SCORM run-time environment is comprised of three elements:
  - *Launch mechanism* - defines a standardized protocol (through the use of a common API) for communication between the delivered learning resources and the LMS.
  - *Application Programming Interface (API)* - used for implementing the communication protocol informing the LMS for the state in which the learning resource can be.
  - *Data model* - a collection of data elements (e.g. the status of a learning resource etc) used to define what information the learning resource and the LMS should communicate.

## 4 Adaptation Approach

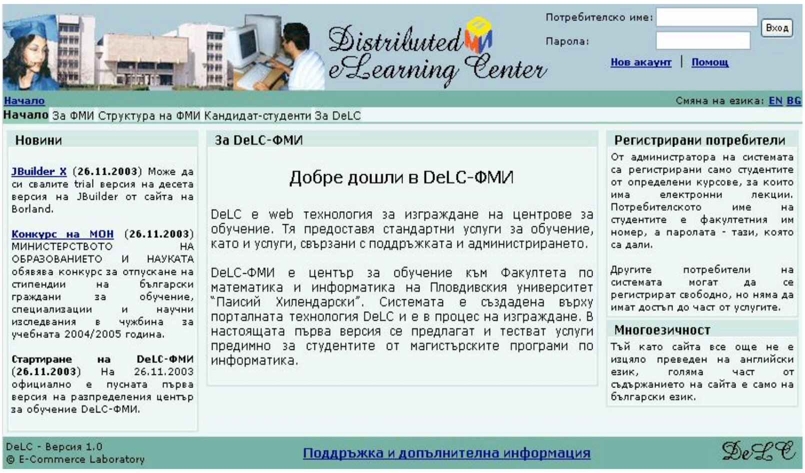
Our approach for adaptation of the DeLC to SCORM includes the following steps:

- Meta-level specification of the DeLC Node – different meta-structures are supported in the current version of the nodes. The structures related to eLearning services are modified in order to satisfy the SCORM meta-level specification.
- Implementation (or adaptation) of the meta-level environment (tools) – the tools that we use to configure the meta-level of the different DeLC nodes are changed in order to generate the new meta-data.
- Implementation (or adaptation) of the meta-level interpreters (run-time) – the run-time modules of the nodes have to be changed so that they can interpret the new structures.
- Realization of the authoring system (educator tools) – according to SCORM specification the learning contents and the learning activities may be created manually or with authoring systems. We have chosen the second approach. We intend to design a complete SCORM-based authoring environment.

- Realization of the sequencing engine (run-time) – to interpret sequencing metadata and execute the required sequencing behaviors during the users' sessions.

## 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 the center (J2EE-based [14, 15, 16]) is in its test phase. The development of the second version of DeLC (SCORM-compliant) is in progress. The main changes are necessary to be made in the client module of the system (Figure 2), which is implemented as an intelligent portal (based on the JetSpeed framework [17]).



**JBuilder X** (**26.11.2003**) Може да си свалите trial версия на десета версия на JBuilder от сайта на Borland.

**Конкурс на МОН** (**26.11.2003**) МИНИСТЕРСТВОТО НА ОБРАЗОВАНИЕТО И НАУКАТА обявява конкурс за отпускане на стипендии на български граждани за обучение, специализации и научни изследвания в чужбина за учебната 2004/2005 година.

**Стартиране на DeLC-ФМИ** (**26.11.2003**) На 26.11.2003 официално е пусната първа версия на разпределения център за обучение DeLC-ФМИ.

Добре дошли в DeLC-ФМИ

DeLC е web технология за изграждане на центрове за обучение. Тя предоставя стандартни услуги за обучение, като и услуги, свързани с поддръжката и администрирането.

DeLC-ФМИ е център за обучение към Факултета по математика и информатика на Пловдивския университет "Паисий Хилендарски". Системата е създадена върху порталната технология DeLC и е в процес на изграждане. В настоящата първа версия се предлагат и тестват услуги предимно за студентите от магистърските програми по информатика.

От администратора на системата са регистрирани само студентите от определени курсове, за които има електронни лекции. Потребителското име на студентите е факултетния им номер, а паролата - тази, която са дали.

Другите потребители на системата могат да се регистрират свободно, но няма да имат достъп до част от услугите.

Тъй като сайта все още не е изцяло преведен на английски език, голяма част от съдържанието на сайта е само на български език.

DeLC - Версия 1.0
© E-Commerce Laboratory

Fig. 2: The facade of the DeLC client node

## 6 References

1. Singh M.P., Vouk, M.A.: Network Computing. In: *Webster, J.G. (ed.): Encyclopedia of Electrical and Electronics Engineering, Vol. 14 (Mu-Nu),* John Wiley & Sons (1999) 114-132.
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, *1st Balkan Conference in Informatics (BCI'2003)*, Thessaloniki, Greece, 21-23 November 2003.
6. Ganchev, I., O'Droma, M., McDonnell, F.: Component-Based Platform for a Virtual University Information System. In: *Mladenov, V. (ed.): Recent Advances in Computers, Computing and Communications. Electrical and Computer Engineering Series of Reference Books and Textbooks*. ISBN 960-8052-62-9. WSEAS Press (2002) 185–190.
7. Hendricks, M., Galbraith, at all: *Professional Java Web Services*, Wrox Press Ltd., 2001
8. Bennett, S.: *Object-oriented systems analysis and design using UML*. 2nd ed. London, McGraw-Hill (2002).
9. OMG *The Unified Modeling Language Specification*, Ver.1.5, www.omg.org/technology/documents/formal/uml.htm (to date).
10. IEEE Learning Technology Standards Committee (IEEE LTSC) (2002), *IEEE Standard for Learning Object Metadata (LOM)*, http://ltsc.ieee.org/wg12/par1484-12-1.html (to date).
11. IMS Global Learning Consortium, Inc. (2001), *IMS Learning Resource Meta-Data XML Binding*, http://www.imsglobal.org/metadata/imsmdv1p2p1/imsmd_bindv1p2p1.html (to date).
12. IMS Global Learning Consortium, Inc. (2003), IMS Content Packaging, http://www.imsglobal.org/content/packaging/index.cfm (to date).
13. Aviation Industry CBT (Computer Based Training) Committee (AICC), *AICC Guidelines and Recommendations (AGR's)*, http://www.aicc.org/pages/down-docs-index.htm (to date).
14. *Java™ 2 Platform*, Enterprise Edition Specification Version 1.4, http://java.sun.com/j2ee/docs.html (to date).
15. *Java™ 2 Platform, Standard Edition, v1.4.2* API Specification (J2SE specification), http://java.sun.com/j2se/1.4/docs/api/index.html (to date).
16. *Java™ 2 Platform,* Enterprise Edition Technical Overview (J2EE Overview), http://java.sun.com/j2ee/white.html (to date).
17. http://jakarta.apache.org/jetspeed/ (to date)