SERVICE – ORIENTED ARCHITECTURE TO SUPPORT LEARNING DESIGN

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

In the paper is proposed developed Service-Oriented Architecture (SOA) to support the processes of managing Learning Design (LD) and their supporting materials, including also repositories and editing tools. The Architecture resuscitates developed Learning Design Module (LDM) and gives it full functionality as integrates it with open source-based Shareable Content Object Reference Model (SCORM)-compliant e-learning system. The design methodology of the SOA is founded on Use Case analysis and visual notation Unified Modeling Language (UML). Realization of the Architecture is grounded on e-learning standards as: Educational Modeling Language (EML), IMS Specifications and SCORM.

Keywords: SOA, Development, IMS LD specification, SCORM.

1. INTRODUCTION

A Unit of Learning (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. and typically contains a LD and learning resources. The LD specifies the workflow in the teaching-learning process [1]. At the abstract level, the LD describes the following process: A person gets a role that entails a set of learning activities for attaining some learning objectives. A method, based on a pedagogical approach, determines which roles get which type of activity at a given time. The learning activities are performed in a learning environment provided with resources and communication facilities. This abstract LD model is implemented in the IMS LD specification that is based on Educational Modeling Language (EML) to create interoperable learning designs [2].

LD requires a substantial supporting framework of components and services if it is to transform the experience of learning technology [3]. LD needs to be integrated in e-learning environment that to offers the creation of e-learning resources and to supports wide variety of services. One approach to joining up modules with enterprises is SOA. It is relatively new approach, but is rapidly gaining popularity because of the low costs of integration coupled with flexibility and ease of configuration. In the paper is proposed developed service-oriented architecture to support the processes of managing LD and their supporting materials, including also repositories and editing tools. The Architecture resuscitates developed LDM and gives it full functionality as integrates it with SCORM – compliant e-learning system.

2. USED E-LEARNING STANDARDS

The development of a framework that supports pedagogical diversity and innovation, while promoting the exchange and interoperability of e-learning

materials, is one of the key challenges in the e-learning industry today. Some standards in e-learning area have been created. The EML developed at the Open University of the Netherlands has served as a basis to the IMS LD Specifications. EML as a modeling strategy for the didactical components allows educators to model a variety of pedagogies [4]. EML allows dealing with the complexity of different educational settings by adding automatically interpreted instructional design features to the exiting e-learning specifications from IMS Global Learning Consortium, Institute of Electrical and Electronics Engineers Learning Technology Standards Committee, etc.

IMS developed LD specification implements the theory of LD into a machine-readable technical standard. It is a highly complex specification, involving several levels. The main reason for implementing a standard for LD is to make digital information encoding learning designs consistent and thus both transportable and re-usable in different software packages.

SCORM aims to foster creation of reusable learning content as "instructional objects" within a common technical framework for computer and web–based learning [5].

SCORM is a model that references a set of interrelated technical specifications and guidelines. It is designed to meet high-level requirements for e-learning content: Accessibility, Adaptability, Affordability, Durability, Interoperability, and Reusability.

3. NEEDS ANALYSIS AND REQUIREMENT DEFINITIONS

In design time of UOL the Author has been planned resources (learning objects) and services which to support the teaching-learning process. The learning objects, described in activity element, typically are linked through to a web page or other content item or are uploaded to content repository. An environment element can contain learning objects and learning services. The location (or URL) of learning objects is known at design time, whereas the location for a learning service is created when a UOL is instantiated. The reason for this is that an LD learning service includes a mapping of LD roles onto the roles in the services. The mapping the roles in the services typically this is done through a management utility provided with the e-learning system in runtime area. The LD is then scanned for all learning services and, with a list of participant for each role, a dedicated instance of the services is set up using the list of participants in the relevant roles and of LD roles to the service roles contained in the UOL's service definition. Setting up the service with the actual participants is done manually. The set up function of the management utility produces a human-readable list of the necessary services together with a list of people mapped to the service's roles. Once a service has been set up, the link (URL) to this service has to be passed back to the runtime area, along with the reference to the service in the LD. From then on, the runtime area can treat a learning service in the same way as a learning object, by simply providing, at the appropriate point, a hyperlink to it in the learner's browser interface (or in instructor's browser for preview). The analysis

of the available architectures supporting LD has done in order to develop appropriate e-learning architecture that integrates LDM.

3.1. Analysis of the available architectures to support Learning Design

The SOA has been chosen, because of the low costs of integration coupled with flexibility and ease of configuration. In SOA the application logic contained in the various systems - such as learner record system, learning environment, directories and so on – are exposed as services, which can then be consumed by other applications. This service layer is interposed between presentation and business logic within typical three-tier architecture (Figure 1).

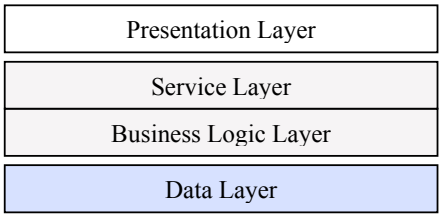
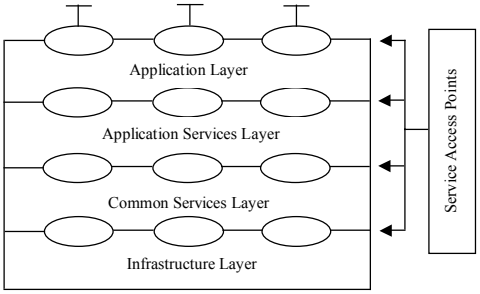


Figure 1 Three-Tier Architecture

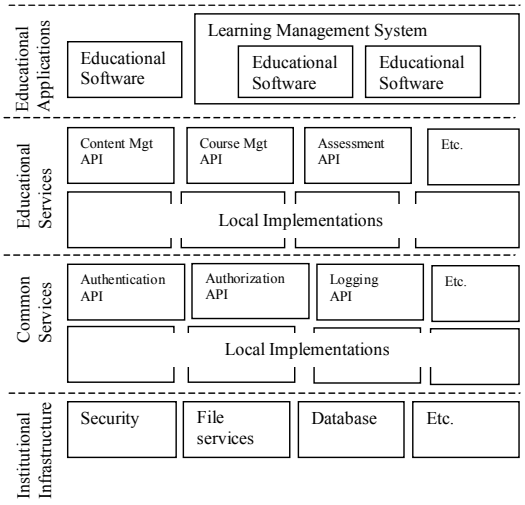
3.1.1. IMS Abstract Framework



The Abstract Framework is document published by the IMS Global Consortium Inc., the body with responsibility for the LD specification [6]. Structurally IMS Abstract Framework consists of four layers: Application, Application Services, Common Services and Infrastructure (Figure 2).

Figure 2 High-level overview of IMS Abstract Framework

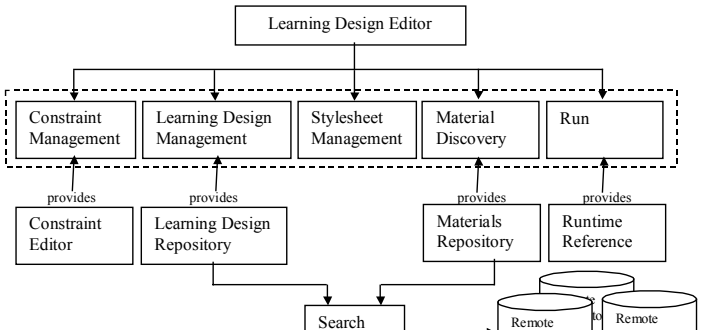
3.1.2. The Open Knowledge Initiative



The Open Knowledge Initiative (OKI) has been developed an architecture framework [7]. Although the OKI model does not define its architecture in terms of web services, but instead as a set of abstract Application Programming Interfaces, there is a lot of commonality between the overall approach taken by OKI and service-oriented architecture. The OKI model defines two large groupings of services: those referred to as Application Services are focused on supporting the needs of educational applications, whether information Management Perspective. The second grouping is called Common Services, and is the set of services associated with access to part of the common technical infrastructure, such as authentication and data management. On top of these two layers of services sit the actual user applications, while beneath them sits the actual infrastructure of the organization – its databases, directories, file systems and so on. This is illustrated in Figure 3.

Figure 3 The OKI Architecture Model

3.1.3. The Valkenburg Group Reference Architecture



The Valkenburg Group is not interested in the internal behavior of the LD editor as such, but the services that need to be place in the wider environment within which the editor is being used [8]. The Valkenburg Group Reference Architecture is presented in the next Figure.

Figure 4 The Valkenburg Group Reference Architecture

This analysis provided a great deal of information about modeling and binding services and components, with an overall goal to synthesize architecture supporting LD. It assisted the identification of the key services needed for integral work of LDM within SCORM-compliant e-learning system.

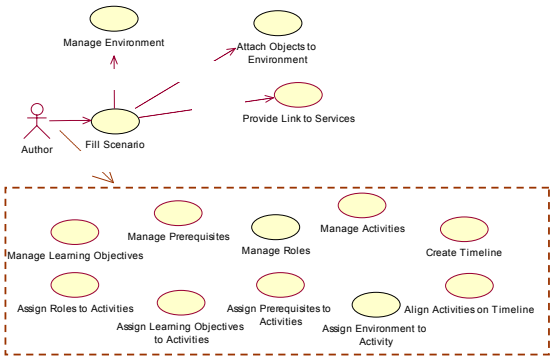
4. DESIGN METHODOLOGY

The design of the SOA supporting LD is based on Use Case analyses and visual notation Unified Modeling Language (UML).

The Use Case technique is an increasingly popular approach to capturing requirements and driving system development. The Use Case Analysis describes the proposed functionality of the new system and represents a discrete unit of interaction between a user (human or machine) and the system. Use Cases are typically related to actors. An actor is a human or machine entity that interacts with the system to perform meaningful work.

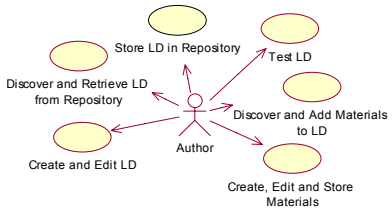
From this functional definition, an abstract model may be developed that sets out in a more formal manner the expected behavior of a realization of the Service, and the structure of the information it deals with. This can be expressed using natural language; however the UML is an extremely useful tool for developing this type of model.

5. USE CASE ANALYSIS



The main Use Case Model of LDM is presented in Figure 5 through UML Use Cases. LDM provides a means of creating pedagogic scenarios, defining the flow of activity as a template, which the author has to populate with resources and services.

Figure 5 UML Use Cases for LDM



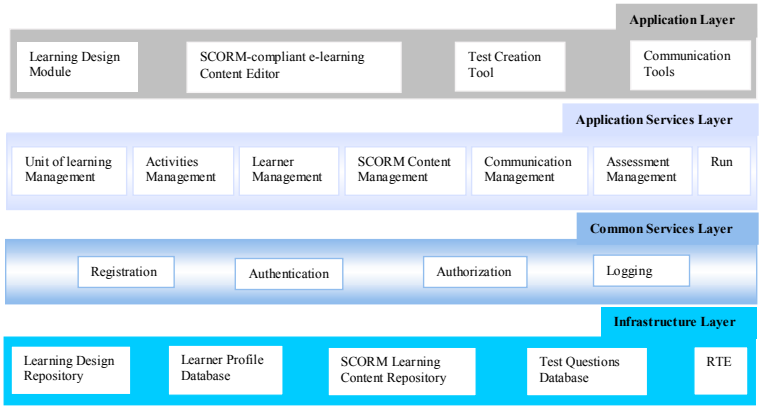
In other hand are examined the kinds of task that Author may need to perform in working with LDM (summarized in Table 1). The main tasks for creating LD are depicted in Figure 6.

Figure 6 UML Use Cases for Main Tasks

Create and edit LD	It is clearly necessary for architecture to support the creation of new LDs, whether from scratch, from an existing LD, or based on a template.
Discover and add materials to LD	It also necessary to incorporate materials within a design, such as learning objects, HTML, images, animations and so on. These materials tend to be created and managed separately from the LD, and then need to be inserted into the LD at the appropriate places. Materials also tend to get stored in repositories that support search mechanisms to enable designers to discover appropriate materials.
Create, Edit and Store Materials	An architecture that supports LD typically need to interact with the workflow for managing materials.
Test LD	The Authors of LD need to test their LD, to step through them and see how it work, and try out the various roles.
Store LD in Repository	The Authors need to store their LD, both in draft form for development, but also to submit finalized design into production XML repository.
Discover and Retrieve LD from Repository	During the creation and editing process, Authors need to find and reuse previous designs, and locate draft learning designs for editing.

Table 1

6. DEVELOPED SERVICE-ORIENTED ARCHITECTURE TO SUPPORT LEARNING DESIGN



In point view of a service - oriented perspective, the services that need to be in place in the wider e-learning environment within the LDM is being used are from main importance.
The proposed architecture model as shown in Figure 7 defines two large groupings

Picture 7 Service-Oriented Architecture Supporting Learning Design

of services: these referred to as Application Services are focused on supporting the needs of educational applications. The second grouping is called Common Services, and is set of services associated with access to parts of the common technical infrastructure. On top of these two layers of services sit the actual user applications, while beneath them sits the infrastructure – databases, repositories and so on. The main functions in each layer are presented in Table 2.

Application Layer	Learning Design Module	LDM provides a range of pedagogical structures, which authors can populate with activities, resources and learners.
	SCORM-compliant e-learning Content Editor	The Editor allows create/edit SCORM e-learning content and learning sequences
	Test Creation Tool	Authors can create automatically scored questions, and randomize test questions.

	Communication Tools	In Communication tools include: Discussion Forums, Internal E-mail, Real-time Chat, and Tool for Announcements
Application Services Layer	Unit of Learning Management	It allows managing of LD and Materials
	Activities Management	Authors can get reports showing the number of times, the time and date on which, and the frequency with which each learner accessed UOL content, discussion forums, and assignments.
	Learner Management	Administrator can manage learner groups or individual learner.
	SCORM Content Management	Authors can selectively release course content based on specific start and end dates. The content packages can be exported to other IMS/SCORM conformant systems, as well as import content from those systems.
	Communication Management	Discussions can be viewed by thread. Posts can include URLs, and can be either plain text or formatted text. There is a chat tool for UOL or group level messaging. Learners can use the internal email feature or instant messaging tool to email individuals.
	Assessment Management	Author can manage questions or test, can manage question's categories
	Run	Run Services allow previewing LDs, SCORM Content and Tests
Common Service Layer	Registration	Learners can self-register. Authors can batch add learners to a course using a delimited text file.
	Authentication	Administrators and authors can protect access to individual courses with a username and password. There is a password reminder option.
	Authorization	Authors can assign limited access to administrative and instructional tools based on pre-defined roles or permissions.
	Logging	Logging is allowed only with unique username and password.
Infrastructure Layer	LD Repository	It stores XML LD files
	Learner Profile Database	The database keeps records for Learners – member's information, test results, groups, forum's subscription and so on
	SCORM Learning Content Repository	It stores SCORM packets
	Test Question Database	The database keeps records for questions, answers and question's categories
	RTE	SCORM RTE provides launch mechanism for SCORM Content and communication mechanism between Content and LMS

Table 2

7. CONCLUSION

SOA is a breakthrough technology that provides an easy, low-cost way for offering of web services. In the paper have been examined and analyzed existing SOA approaches to support LD. Use Case models make clear the functions of LDM and contribute to full integration of LD in SCORM-compliant e-learning environment. The SOA has been developed as a useful framework for identifying and specifying the components of architecture for authoring and managing LD.

8. REFERENCES

[1] CETIS, *Learning Design inspiration*, 2003, http://www.cetis.ac.uk/content2/

[2] IMS Global Learning Consortium, Inc., *IMS Learning Design Information Model, XML Binding Document, Best Practice Guide*, 2003

[3] Koper R., C. Tattersall, *Learning Design*, Springer, 2005

[4] Koper, R., *Modeling units of study from a pedagogical perspective: The pedagogical meta-model*, 2001

[5] *ADL Shareable Content Object Reference Model*, July 2, 2004, www.adlnet.org

[6] *AF IMS Abstract Framework*, 22 January, 2004, www.imsglobal.org

[7] Open Knowledge Initiative, *Open service Interface definitions*, July 2, 2004, web.mit.edu/oki

[8] Koper R., *Combining reusable learning resources and services to pedagogical purposeful units of learning,* Kogan Page, London, pp 44-59, 2003