E-learning portal based on the IMS consortium specifications

JAMAL-EDDINE EL KHAMLICHI, JEAN-PIERRE PECUCHET, FRANCOISE GUEGOT
Laboratoire LITIS
INSA de ROUEN
BP 08, Place Emile Blondel 76131Mont Saint Aignan Cedex
FRANCE
{jamal.elkhamlichi, Jean-Pierre.Pecuchet, francoise.guegot}@insa-rouen.fr

Abstract: - This paper trait the LU (Learning Units) reuse and the interoperability between tools and software services. It presents a portal of e-learning, based on the IMS consortium specifications and on the web services technologies. It uses MOT and RELOAD tools to graphically design and to produce the LU. It consists also of a LMS, used to manage and to organize the learning. This LMS is conceived to be opened. It bases itself on the web services technologies which allow its interoperability with other tools or web services. It also integrates a catalogue, used to search and to download the LU from the LU repository.

Key-Words: - e-learning, reuse, interoperability, IMS, LMS, web services, learning unit, educational scenario.

1 Introduction

The work presented in this paper is situated into an e-learning context. The main characteristic of this context is the emergence of the use of the ICT (Information and Communication Technologies) for contents production, diffusion and consultation. The generalization of the use of ICT in e-learning, leads to an explosion of LU on the internet. Indeed, many studies [15; 16; 22] count hundreds of LMS (Learning Management System) able to provide the LU for e-learning. In many cases these LU are not reusable by other LMS. In fact, it is necessary to develop specifics UL for most LMS. By consequence the production cost of LU is high.

To answer to this problematic, in this last decade, two approaches have appeared to treat the problem of LU reuse [17]. The first approach is interested by the reuse of the LOs (Learning objects). The aim is to create repositories of LOs shared on internet. ARIADNE, COLIS, Edusource, DLESE and MERLOT are examples of projects which treat the reuse of LOs. The second approach is interested by the reuse of the educational scenario. Educational languages appeared as IMS (Instructional Management Systems), EML (Educational Modeling Language) and MISA (Méthode d'Ingénierie des Systèmes d'Apprentissage). They propose models which treat design and reuse of educational scenarios. Our work is inscribed into this last approach. We use IMS as modeling language for the reasons which we shall evoke further in this paper.

Also, the emergence of the ICT leads to an explosion of the computer tools (forum, chat, LMS, etc.) and services. Very often, these tools and services are not interoperable. The web services technologies represent a solution allowing the interoperability.

The portal which we elaborate uses MOT and RELOAD tools for LU design and production. It also consists of a LMS, constituted of a set of web services. A catalogue is also in process of development to share the LU being in local or distant LU repositories.

In this paper, we present first of all the notion of LU. Then, we detail the portal functional architecture. Finally, we focus on the various tools and the services which compose the portal, which are respectively: MOT, RELOAD, LMS, catalogue and LU repositories.

2 Learning unit

The modeling language, proposed by the IMS consortium, is widely inspired from R. Koper works [10]. 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 script every type of LU (e.g. lessons, problem based learning, QCM, etc.).

An LU is introduced by IMS as an abstract term which makes reference to an element of learning or of education as for example a lesson, a module or a QCM [6]. 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 modeled 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 [3; 4; 7]:

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

The first child is an optional element, called Metadata. It describes the manifest as a whole. For that, we use the IEEE-LOM [12] metadata scheme.

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 <learning-design> element. This last one contains the elements which describe the educational scenario. Without describing in detail this element, we remind that it summarizes the idea according to which the educational scenario takes place as a theater 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 has to have to realize the activity. The educational scenario is organized in A, B and C three levels [6]. 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.

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 physical files are the local and external files making up the contents of a LU. They represent electronic representations of media, such as text, sound, images, animations, graphs or any piece of data that can be rendered by internet and presented to the learners. Each of these media can have a multiple forms of representation. For example, a sound can have the following format WAV, MP3, MPC. 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. This last is a concise web delivery zip format. The use of a concise zip format facilitates the transport of the PIF file on internet.

3 Portal functional architecture

In most of e-learning portals (e.g. INES, WebCT, Explor@), the authors produce specific LU for every LMS. In such situation, an LU cannot be reused by other LMS and in different contexts. Also, mostly the services and tools proposed by these portals are not interoperable. They are strongly coupled and no offer the flexibility at the level of the interoperability.

The remedy of these two problems implies the emergence of new actors, tools and services. We propose in follows the portal functional architecture (Fig.1) which takes into account both problems of LU reuse and tools and services interoperability. The elements of this architecture are:

The *actors* represent the persons who play the various administrative, educational, technical roles.

The *LMS* is the learning management and the organization system: management of the learners, individualization of the learning, evaluation of the learners, etc.

The *CMS* (Content Management System) is the LU management system. It helps to create, to updates and to manage the LU. These systems base themselves on two principles. The first one 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 the LMS and the 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.

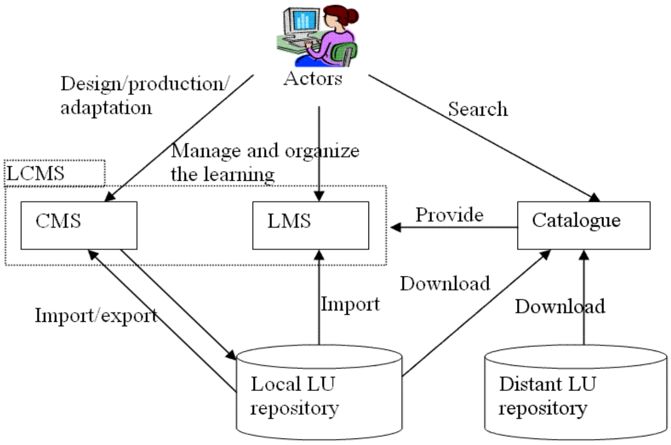


Fig.1: Portal functional architecture.

4 CMS

These last years several editing tools (ALFANET, ASK-LDT, Collage, COSMOS, CopperAuthor, Elive LD-suite, Educreator, Eduplone, MOT, RELOAD, netUniversité, SCOPE Library, TheCoDe), appeared [23]. The features and the characteristics of these tools differ from a tool to the other tool. We chose to use MOT and RELOAD tools for the following reasons:

1. RELOAD and netUniversité is the only tools which allow producing the A, B and C three levels of the IMS consortium specifications. RELOAD offers also the possibility of importing and exporting the LU.

2. MOT is a tool which allows to graphically designing the educational scenarios. It offers the following advantages [2; 18]:

- Illustrate the relations of the educational scenario elements.
- Makes evident the complexity of the interactions between the actors.
- Facilitate the communication of the educational scenario.
- Assures the perfection of the educational scenario.
- Help on examination of the educational scenario by minimizing the use of texts.

3. Both tools are freely downloadable.

The educational design is supported by the MOT tool. Without describing in detail this tool, we remind that this last one proposes a set of graphical notations. These last ones represent four types of knowledge [13; 19; 20; 21; 24]:

1. The procedures which allow constituting a set of activities and of connection relations between them in a more or less complex way.

2. The concepts which constitute a set of objects, of events and of symbols which share common characteristics.

3. The principles which are relations of cause with effect in a process. They are the roles or the rules.

4. The facts which represent the instances of the previous three abstracts knowledge.

The terms of the educational scenario are classified according to these four types of knowledge. For example, an activity in IMS is a procedure in MOT. These terms can be connected by six types of links: instantiation, composition, precedence, input/product, regulation, application. For example, a link of precedence can connect two activities; this is means that one of both activities precedes the other one.

We also remind that MOT proposes two predefined graphical models. The first one is the main model. It allows describing the method, the central element of the educational scenario. The second is sub model. It allows describing acts specified in the main model. The elements which constitute the principal model are the first ones to specify by the designer (Fig.2). But, for every act created, the designer can describe the corresponding sub model.

After the design of the educational scenario, MOT produces in zip format, the A level of the educational scenario. Works to cover two other B and C levels will be making in LORNET project [11].

The zip format, generated by the MOT tool, facilitates its import in the RELOAD tool (Reusable ELearning Object Authoring and Delivery), to edit the B and C levels [1].

Both MOT and RELOAD tools are intended to designers mastering the terminology proposed by IMS consortium. To remedy to this constraint, an approach of iterative and participative design in the form of educational design tasks is elaborated to guide the designers [4]. This approach can be integrated into the tool MOT as a layer of predefined graphical tasks models having a superior level to the both models presented previously.

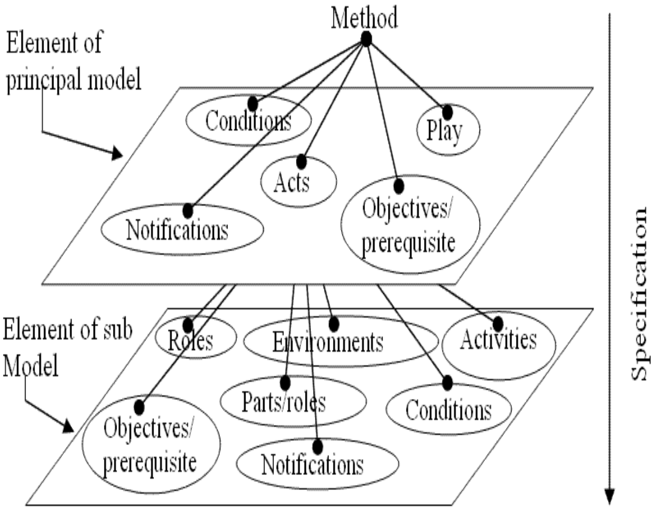


Fig.2: specification direction of the educational scenario elements in MOT tool.

5 LMS

LMS proposes a set of tools and of services for the management and the organization of the learning. To facilitate the interoperability between these tools and these services on one hand and between them and the others tools and services of the portal on other hand; we use the web services technologies.

5.1 LMS Software architecture

The LMS is developed according to the three tiers architecture. It come from J2EE (Java 2 Platform, Enterprise Edition) specifications [9]. The three tiers are (Fig.3):

1. The presentation tier which represents the HMI (Human Machine Interface) of the LMS. It is developed in JSP (Java Server Pages) programming language.
2. The business tier is constituted by the business logic of the LMS. It integrates web services and APIs (JDBC, JavaMail, Java-WS, etc.), proposed by J2EE.
3. The data tier is constituted by data bases in MSQL SGBDR (Système de Gestion de Base de Données Relationnelles) containing the data necessary for the application.

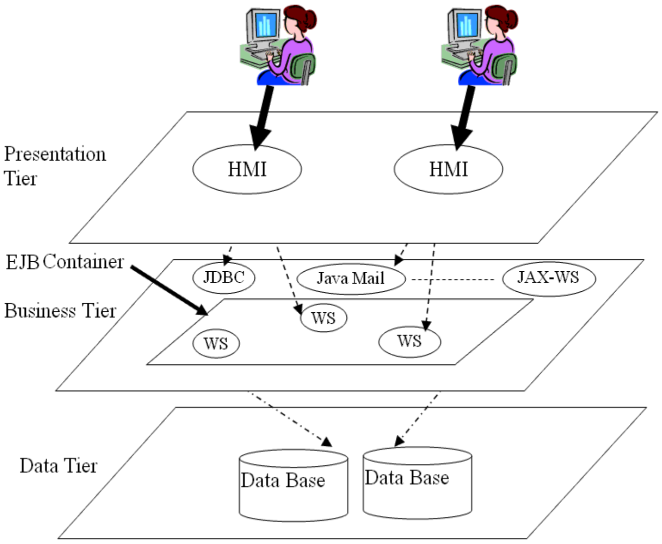


Fig.3: the three tiers LMS software architecture.

There are two types of web services. The first one is called application web services. These last ones provide the functionalities of learning, administration, communication, etc. However, the second type is called common web services. They are in the form of a set of web services available for the application web services. A common web service can use the other common web services. However, a common web service is available for all others web services.

This architecture resolves the problems relative to the performances, to the network traffic and to the maintenance, encountered with the two tiers architecture (client/server). Also, the conception of the business logic in the form of web services offers the possibility of reusing the components of the LMS and of evolving of the LMS itself.

5.2 Web service software architecture

We use the EJB (Entreprise JavaBeans) to implement the web services [5]. Every web service consists of the following five tiers (Fig.4):

1. The first one contains the SOAP interfaces. They interface the tier "façade client" by "Axis SOAP toolkit for Java".
2. The second is called "façade client". It consists of "business delegates" which implement "native Java interface" for components EJB. They contain the necessary code to facilitate the connection to the EJB interfaces.
3. The third consists of beans sessions. They deal with all the calls for the business logic. For example, a call for the preparation of an educational scenario instance.
4. The fourth represents the business logic. This last one contains all packages representing the applicative logic of the web service.
5. The last one is the Data Access tier. It is responsible for all the direct interactions with data bases. It consists of five packages: Dossier, Run, UoL (Unit of Learning), User and Role. The Dossier package is the major component of this tier.

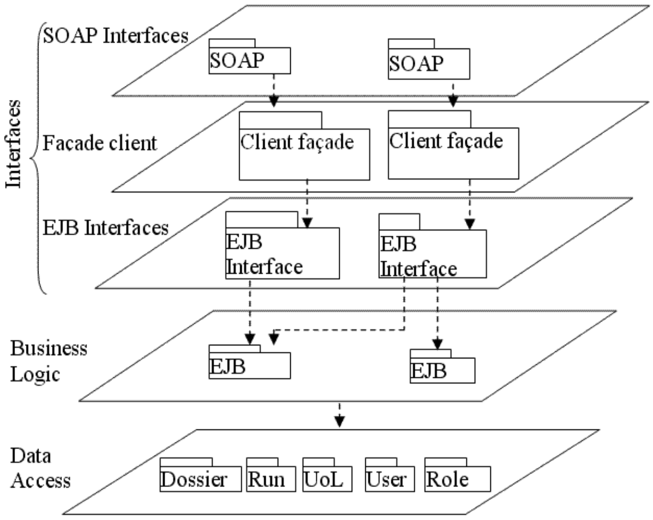


Fig.4: the web service software architecture.

5.3 Communication between web services

The interactions between the web services are made as we can see it on the figure below (Fig.5). There is a communication canal which is responsible of the interoperability between the web services. It provides an infrastructure of communication web services. We distinguish three categories of interaction interfaces which are supported by the communication canal [8]:

1. The application web services interface (A1) is used to allow the interoperability between the application web services.

2. The common web services interface (A2) is used to allow the interoperability with the common web services which are available for the application web services.

3. The run-time interface is used to interconnect an actor who interacts with the HMI and the run-time environment.

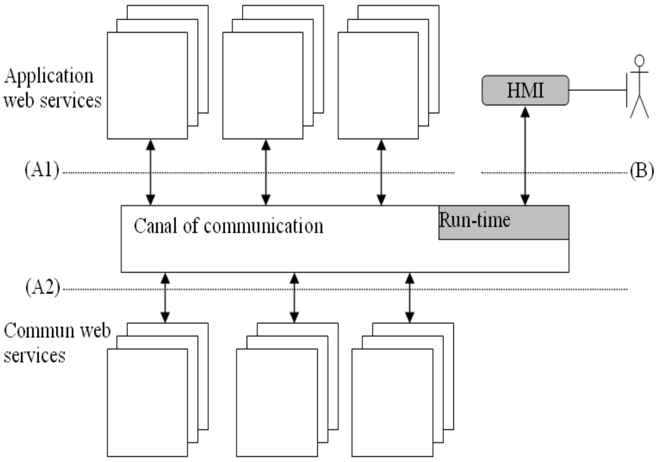


Fig.5: the three interaction interfaces between the web services.

The access to every web service is made through a SAP (Service Access Point). Every web service possesses a single SAP. Several web services can be combined to form a component offering a particular service. Contrary to a web service, a component can support several SAP.

The communication canal is at the same time software application and physical equipment. It consists mainly of SOAP (Simple Object Access Protocol) messages (Fig.6). Other protocols (http, HTTPS, FTP, SMTP) have in responsibility the transport of the SOAP messages contents on a physical network (LAN / WAN, ATM, wireless, etc.).

A SOAP message is the applicative communication way between the component which asks the service and the component which provide it. Both components can interpret the SOAP message. A message can be synchronous. In this case the answer is immediate. It can be also asynchronous. The answer can be obtained after some times. There is also a polled message. For this last one, the service requester has to authorize sends of the answer by the service provider, before receiving it. The type of implemented message depends on the wanted service. For example, an authentication service has to receive answers to its requests in synchronous mode to allow the user the access to the LMS.

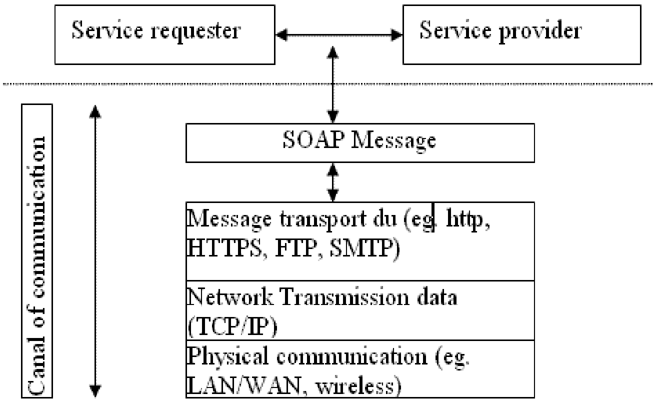

Service requester
Service provider
SOAP Message
Message transport du (eg. http, HTTPS, FTP, SMTP)
Network Transmission data (TCP/IP)
Physical communication (eg. LAN/WAN, wireless)
Canal of communication

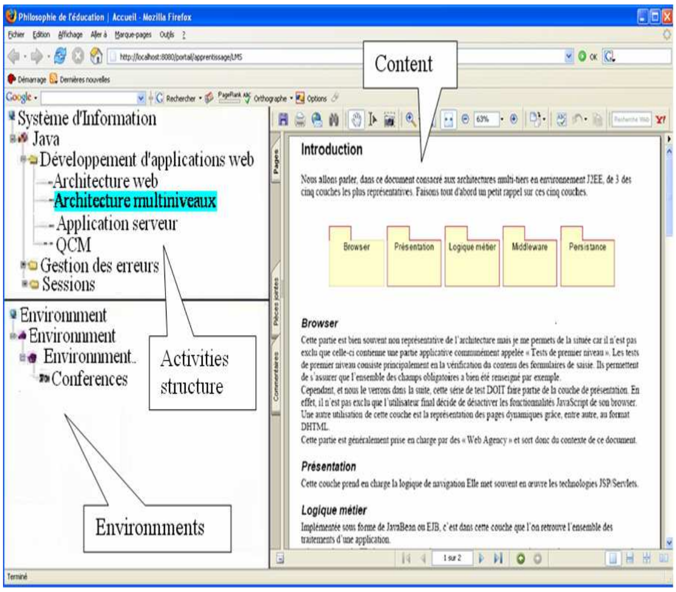
Fig.6: the communication canal elements.

5.4 CopperCore : example of web service

There are in the literature several educational scenarios players (SLeD, Edubox, netUniversité, RELOAD, CopperCore). SLeD and RELOAD bases itself on CopperCore. This last one represents the first player available freely on internet. NetUniversité and Edubox are owners.

CopperCore supports the A, B and C three levels of the educational scenario [14]. It is developed in the OUNL (Open University of NetherLands). It is in the form of a web service and open source. The LMS integrates CopperCore as educational scenarios player.

The figure below (Fig.7) shows the result of execution of the Java lesson by the LMS. The screen is constituted by three main parts. The superior left part represents the activities structure. The lower left part shows the environments of the selected activity. The right part shows the contents of the selected activity or the selected environment.



Introduction

Nous allons parler, dans ce document consacré aux architectures multi-tiers en environnement J2EE, de 3 des cinq couches les plus représentatives. Faisons tout d'abord un petit rappel sur ces cinq couches.

Browser

Cette partie est bien souvent non représentative de l'architecture mais je me permets de la située car il n'est pas exclu que celle-ci contienne une partie applicative communément appelée « Tests de premier niveau ». Les tests de premier niveau consiste principalement en la vérification du contenu des formulaires de saisie. Ils permettent de s'assurer que l'ensemble des champs obligatoires a bien été renseigné par exemple.
Cependant, et nous le verrons dans la suite, cette série de test DOIT faire partie de la couche de présentation. En effet, il n'est pas exclu que l'utilisateur final décide de désactiver les fonctionnalités JavaScript de son browser. Une autre utilisation de cette couche est la représentation des pages dynamiques grâce, entre autre, au format DHTML.
Cette partie est généralement prise en charge par des « Web Agency » et sort donc du contexte de ce document.

Présentation

Cette couche prend en charge la logique de navigation Elle met souvent en œuvre les technologies JSP/Servlets.

Logique métier

Implémentée sous forme de JavaBean ou EJB, c'est dans cette couche que l'on retrouve l'ensemble des traitements d'une application.

Fig.7: example of LMS screen.

Two other web services are also operational: the forum of asynchronous discussion and a service of learner profiles management. These two web services were conceived and developed according to the web service architecture presented previously.

6 Catalogue and LU repository

There are two types of LU repositories: local and distant. The local repository is a data base which is on the portal server. It contains the LU created by actors. The LU which are in this data base are importable to use them or modify them by the LMS, the CMS or the catalogue. The distant LU repository is a data base which contains LU which are on another server than the portal server. The LU which in this data base are downloadable only by the catalogue.

Every LU repository offers some number of services (e.g. search, delivery, stocking, etc.), implemented in the form of web services. These web services allow the interoperability of repository with the other tools (LMS, CMS or catalogue).

7 Conclusion

The work presented in this paper is inscribed within the research works having as objective the realization of a method to design and to produce the reusable LU. This method is constituted by three major components. The first one is in the form of two models describing the LU. While the second, represents an approach of design of the educational scenarios. Finally, the third is in the form of a portal offering a set of tools and of services for the design, the production, the execution and the search of the LU. It is this last component which was the object of this paper. It was realized by taking into account the reuse of the LU and the interoperability between tools and services.

Our future work will focus on the elaboration of a catalogue, which will allow to search and to download the LU being on local or distant repository. The search will be made according to criteria such as the educational objectives and the prerequisites. The IMS consortium proposes the specifications RDCEO (Reusable Definition of Competency or Educational Objective), in the form of an information model. This last one provides a formal representation of the knowledge and the competence. Our objective will be to represent the educational objectives and the prerequisites in the form of ontologies by basing itself on the specifications of RDCEO. The implementation of these ontologies by the catalogue will allow obtaining a better result of search of the LU, with regard to a keyword search.

References:

[1] D.-M. Colin; P. Beauvoir and P. Sharples, Learning Design based on Graphical Knowledge-Modelling, *In Proceedings of the UNFOLD/Prolearn workshop,*. Heerlen : R. Koper; C. Tattersall and D. Burgos, pp. 317-323, 2005.

[2] D.-F. Dansereau, *The development of a learning strategies curriculum,* New York: Academic Press, 1978.

[3] J.-E. Elkhamlichi; F. Guegot and J.-P. Pécuchet, Design and generation of Collective Educational Activities. *In Proceedings of the 5th IEEE International Conference on Advanced Learning Technologies*, Kaohsiung: IEEE Computer Society Press, pp. 293-294, 2005.

[4] J.-E. Elkhamlichi; F. Guegot and J.-P. Pécuchet, *Methodology for Design and produce the reuse content*. Forthcoming in IFIP conference, 2006.

[5] Official web page of Sun Developer Network. *http://java.sun.com/products/ejb/*, 2005

[6] IMS Learning Design Information Model, *http://www.imsglobal.org/learningdesign/ldv1p0/imsld_infov1p0.html,* 2003.

[7] IMS Content Packaging Information Model, *http://www.imsglobal.org/content/packaging/cpv1p2pd/imscp_infov1p2pd.html*, 2005.

[8] IMS General Web Services Base Profiles, *http://www.imsglobal.org/content/packaging/cpv1p2pd/imscp_infov1p2pd.html*, 2005.

[9] Official web page of Sun Microsystems J2EE, *http://java.sun.com/j2ee/index.jsp*, 2005.

[10] R. Koper, *Modeling Units of Study from a Pedagogical Perspective, the pedagogical meta-model behind EML, http://eml.ou.nl/introduction/docs/ped-metadodel.pdf*, 2005.

[11] Web Page of LORNET project, *http://www.lornet.org/index.htm*, 2006.

[12] Official web page for LOM metadata, *http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf,* 2004.

[13] M.-D. Merrill, *Principles of Instructional Design,* New Jersey: Educational Technology Publications, 1994.

[14] P. McAndrew; R. Nadolski and A. Little, Developing an approach for Learning Design Players, *In Proceedings of the workshop UNFOLD/Prolearn*, Heerlen: R. Koper; C. Tattersall and D. Burgos, 2005, pp. 351-358.

[15] Oravep Comparative study, *http://www.educnet.education.fr/superieur/plateforme.htm*, 2000.

[16] Web page containing the open source LMS, *https://fnl.ch/LOBs/LOs_Public/OpenSourcePlatf.html*, 2005.

[17] J.-P. Pernin and A. Lejeune, Dispositifs d'apprentissage instrumentes par les technologies: vers une ingénierie centrée sur les scénarios, *In Proceedings of the TICE 2004 conference*, UTC, Compiègne: TICE 2004 conference, 2004, pp. 407-414.

[18] G. Paquette; M. Léonard; K. Lundgren Cayrol; S. Mihai la and D. Gareau, Learning Design based on Graphical Knowledge-Modelling, *In Proceedings of the workshop UNFOLD/Prolearn*, Heerlen: R. Koper; C. Tattersall; and D. Burgos, 2005, pp. 1-20.

[19] G. Paquette, La modélisation par objets typés : une méthode de représentation pour les systèmes d'apprentissage et d'aide à la tâche, *Sciences et techniques éducatives,* 3, 1, 1996, pp. 9-42.

[20] A.-J. Romiszowski, *Designing Instructional Systems*, New York: Nichols Publishing, 1981.

[21] R.-D. Tennyson, Cognitive Learning Theory Linked to Instructional Theory, *Journal of Structural Learning,* 10, 1990, pp. 249-258.

[22] Thot official web page, *http://thot.cursus.edu/*, 2005.

[23] Web page of unfold project, *http://www.unfold-project.net/general_resources_folder/tools/currenttools*, 2005.

[24] C.-K West; J.-A. Farmer and P.-M. Wolff, *Instructional Design, Implications from Cognitive Science*. Boston: Allyn and Bacon, 1991.