Metadata Interoperability and Ingestion of Learning Resources into a Modern LMS

Aikaterini K. Kalou[1], Dimitrios A. Koutsomitropoulos[1], Georgia D. Solomou[1] and Sotirios D. Botsios[2]

[1] High Performance Information Systems Laboratory (HPCLab),
Computer Engineering and Informatics Dpt., School of Engineering,
University of Patras, Building B, 26500 Patras-Rio, Greece
{kaloukat, kotsomit, solomou}@hpclab.ceid.upatras.gr
[2] Dataverse Ltd,
98 G. Papandreou Str., 54655, Kalamaria, Thessaloniki, Greece
sdm@dataverse.gr

Abstract. Over time, academic and research institutions worldwide have commenced the transformation of their digital libraries into a more structured concept, the Learning Object Repository (LOR) that enables the educators to share, manage and use educational resources much more effectively. The key point of LORs interoperability and scalability is without doubt the various standards and protocols such as LOM, SCORM etc. On the other hand, Learning Management Systems have boosted the expansion of the e-learning notion by providing the chance to follow remotely courses of the most well-known universities. However, there is no a uniform way to integrate these two achievements of e-learning and assure an effective collaboration between them. In this paper, we propose a solution on how we can ingest learning objects metadata into the Open eClass platform.

1 Introduction

Thousands of academic and research institutions worldwide implement Learning Object Repositories (LORs) [11,13] in order to maintain and manage their intellectual outcome, with several of them allowing open and free access to their content. A Learning Management System (LMS) on the other hand is a software package that can be used to administer one or more courses to one or more learners [1]. An LMS is typically a web-based system that delivers and manages instructional content, identifies and assesses individual and organizational learning or training goals, tracks the progress, and collects and presents data for supervising the learning process [16].

Modern LMSs offer an abundance of features for course design, creation, management and administration on-line. As a result they concentrate rich learning material or learning objects that can be reused and made available for other purposes. These learning objects most often originate from the instructor's manual labor and their curation level heavily depends on his personal authority. The availability of external learning resources managed and curated by LORs could considerably alleviate the burden of providing additional material to students.

Interoperability between educational systems has been investigated before, either by questing for a common unifying model, proposing metadata mappings [8] or even by specifying bridging languages [3]. Indexing of LMS material into a LOR has been also proposed [14]. However, most LMSs do not yet support automated ingestion of LOR material and when they do, the approach is mostly fragmented and oriented towards specific repositories only. Key reasons for this situation seem to include interoperability concerns and metadata schemas incompatibilities. For example, there exists a custom plugin for the Moodle LMS [10], which allows loading of objects from the MERLOT learning repository [12], through a proprietary process. Next, the EU-funded LUISA project [2] created an infrastructure that supports the integration of LOR with a Learning Content Management System (LCMS). It mainly addressed the key issue of Digital Rights Management (DRM) interoperability by exploiting semantic technologies.

Therefore, in this paper we propose a method that addresses the challenge of ingesting external educational resources into a modern LMS. Our proof-of-concept comes from a prototype implementation on top of Open eClass, a widely used LMS by higher education institutions worldwide. First, we consider the internal metadata schema imposed by eClass and examine how it maps to the well-known LOM standard (Section 2). This is necessary in order to assess the interoperability potential of the application and to identify possible points of alignment (mappings) with external collections' metadata. Then we present the design and implementation of our approach and specify a procedure that could automate the mapping process with external LORs (Section 3). Section 4 exemplifies our contribution through a possible usage scenario that involves the ingestion and manipulation of a thematic collection of learning material by the instructor, using the application's front-end. Finally, Section 5 summarizes our conclusions and future work.

2 Metadata Interoperability in Open eClass

Open eClass [4] is a free and flexible e-learning platform which can address the asynchronous distance learning demands of higher education institutions. In addition, it is the solution offered by the Greek Academic Network GUnet to support asynchronous e-learning services in universities. It is mainly designed, developed and supported by the GUnet Asynchronous eLearning Group.

However, it is not clear which of the standards and protocols the eClass platform embeds in its nature. In this section, we make an attempt to record the metadata schema of the eClass platform, map it to the LOM standard and identify which other specifications and standards are included in its infrastructure. We also explain our mapping strategy with external repositories.

2.1 Mapping to LOM and other standards

The IEEE Learning Object Metadata (LOM) [6] is a widely adopted standard aiming at the description of educational material (learning material) and training resources (learning resources). The LOM conceptual schema defines the structure of a metadata instance for a learning object (LO). Specifically, LOM contains over sixty elements which are further classified in nine categories (see Figure 1) and each one of them contains metadata about various aspects of a LO. The categories of LOM at the top of the data hierarchy are *General, Life-cycle, Meta-metadata, Technical, Educational, Rights, Relation, Annotation*, and *Classification*.

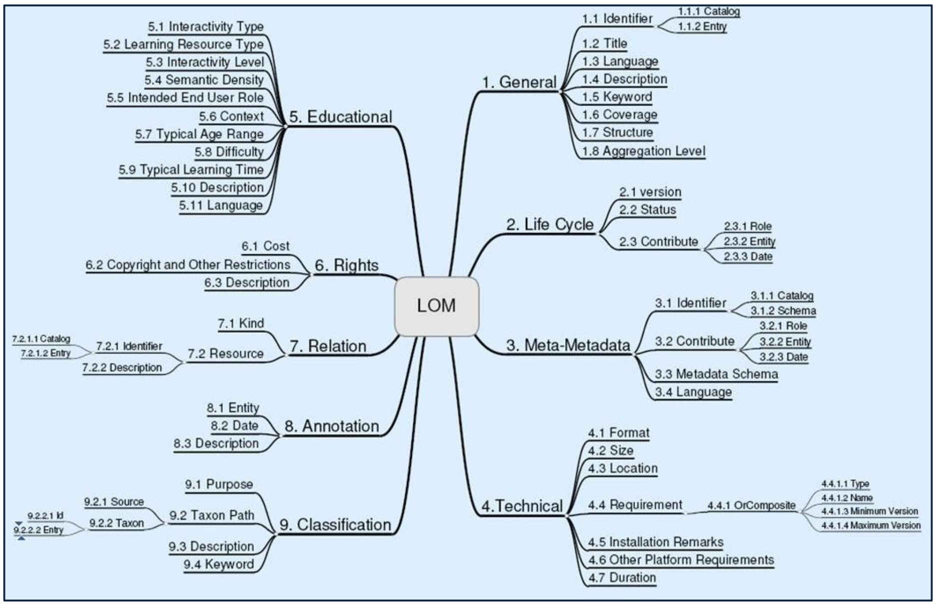

Fig. 1. LOM standard: The hierarchy of elements [7]

The LOM standard respects the general granularity hierarchy of LOs, containing the following six aggregation levels [15]. Similarly, the eClass platform seems to partially adopt these levels:

- *1st Level*: Curriculum
- *2nd Level*: Course
- *3rd Level*: Unit
- *4^{rth} Level*: Topic
- *5th Level*: Lesson
- *6th Level*: Fragment

Based on the above aggregation levels and the set of available metadata, we made an attempt to find out intuitively the common elements between Open eClass and LOM. Our initiative is based on the native structure of Open eClass that appears to be compatible with the LOM specifications [5]. Tables 1-4 gather some of the metadata that can be defined for the wider notion of a course in the context of the Open eClass environment.

Table 1. Mapping of Open eClass metadata to LOM standard - Course

Open eClass	LOM standard
Title	General.Title
Code	General.Identifier.Entry
yearOfStudy	General.Coverage
Semester	
Type	Educational. Learning Resource Type
Language	General. Language
targetGroup	Educational.Intended End User Role
Description	General. Description
Keywords	General.Keyword
eudoxusCode	General.Identifier.Catalog
License	Rights.Copyright and Other Restrictions

Table 2. Mapping of Open eClass metadata to LOM standard - Curriculum

Open eClass	LOM standard
curriculumTitle	General.Title
curriculumDescription	General. Description
curriculumKeywords	General.Keyword
curriculumTargetGroup	Educational.Intended End User Role

Table 3. Mapping of Open eClass metadata to LOM standard – Unit

Open eClass	LOM standard
Title	General.Title
Description	General.Description
Keywords	General.Keyword

Table 4. Mapping of Open eClass metadata to LOM standard – Fragment/Material

Open eClass	LOM standard
Digital	
digital-url	General.Identifier
digital-library	General.Description
Multimedia	
multimedia-title	General.Title
multimedia-description	General.Description
multimedia-keywords	General.Keyword

Link	
link-title	General.Title
link-Category	Classification.Keyword
link-Description	General.Description
link-URL	General.Identifier

Note at this point that most of the metadata of the structural units *Course* and *Curriculum* should be completed during the creation of the Course by the head instructor. On the other hand, metadata related to *Units* and *Fragment-Material* can be filled out once the instructor enriches the *Course* with educational materials categorizing them in conceptual units.

Besides, the Open eClass qualifies to participate in popular international aggregators and directories, thus contributing to the increasing visibility of each educational repository to the broader public. The interconnection with other services is achieved by implementing the OAI-PMH protocol (Open Archives Initiative Protocol for Metadata Harvesting) [9] and the support for simple Dublin Core schema.

2.2 Mapping external collections

In the context of Open eClass, a *course link* is considered a course fragment (Table 4) and can be specified by setting the following three metadata, a *URL*, a *URL Title* and a *URL Description*. *Category* is optional and can be used to provide an arbitrary header to group links, e.g. in a thematic manner. In our approach, we should extract all these useful and essential details from an incoming collection of external learning material and then map them to one unified metadata schema, which contains the above three fields at minimum. The rest of the metadata annotations are not lost. Rather, it would be easy to retrieve them from their sources directly using their unique URL or harvest them through an OAI service provider. The three metadata elements pertain to not the course itself but only to the external links a course may point to. Therefore, external learning material is mostly referenced, rather than replicated within the LMS. Such a strategy would only put unnecessary burden to the LMS database and would be hard to maintain or keep up-to-date.

In addition, incoming collections may contain metadata in proprietary structure. For example, in openarchives.gr, the main result node of the response is identified with the `<entry>` node and roughly follows RSS. For the Europe PubMed Central, each `<result>` node corresponds to a search result and so on. Therefore, in the next section, we also discuss a process for aligning external collections to our schema.

3 Importing Learning Objects

In this section, we present how we extend the Open eClass infrastructure and the overall design of our application and its interaction with learning object collections available within LORs. We also describe thoroughly the main features that have been implemented and advance the user experience in the Open eClass platform. Finally, we illustrate the alignment procedure that may be followed so as to amalgamate external learning objects into the Open eClass metadata schema.

3.1 Design and Architecture

The modular philosophy of the Open eClass, as a typical and complete Learning Management System, allows us to extend its capabilities with ready-made modules and to reuse them in order to fulfill our new needs and purposes. To this end, we utilize the *Link module* that offers a front-end to enhance a course with useful resources and then group them together under categories, which the instructor can also define. The main obstacle of this module is that it allows instructors to add just one link each time.

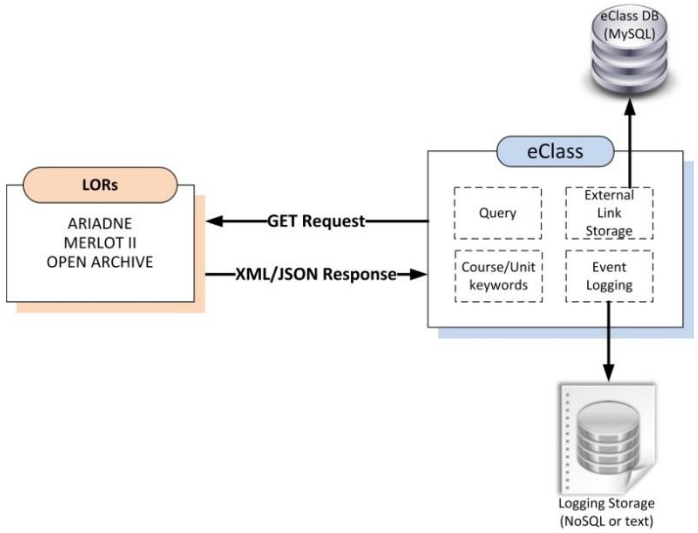

Fig. 2. Architecture for ingesting collections of learning objects in Open eClass

In order to overcome this limit, we have enhanced this module so as to support a more generic, batch way to add external links. In our case, the external links are not just links to web pages, but they are learning objects with high-value educational content. More precisely, the instructor can now add external links through the interface of the *Link* module by communicating with services that harvest LORs, such as Openarchives.gr, ARIADNE and MERLOT II. Additionally, the instructor can load directly a collection of learning object metadata, in XML format, available on the Web or using an appropriate service. The architecture of our re-engineered module is illustrated in Figure 2.

3.2 Implementation and Features

The modification of the *Link* module involves not only the front-end but also the back-end infrastructure. In order to assure asynchronous communication between the back-end and front-end, the capabilities of AJAX technology have been leveraged. All the new features are summarized below:

– *Front-end of the Link module*: The interface of the *Link* module is modified so as to display the learning objects metadata that are loaded from external collections. More precisely, after completion of the loading process, the web interface presents a table of learning object metadata organized into categories. These categories are in fact the keywords used in the query process (see function *Communicating with external web services* below). For each learning object, the URL, the title and the description are available to the end-user. All the front-end functionalities are implemented using the Javascript JQuery library, which is already exploited by the existing module.

– *Metadata retrieval for a course*: In order to discover new learning objects we are based on the existing metadata of the course. In particular, the keywords of the course, the course title and the title of each unit (in case the course includes those) should be recovered. The above function requires communication with the central database of the Open eClass (MySQL, PHP).

– *Link and category storage*: All the learning objects metadata and their corresponding categories, which the instructor has selected from the front-end, will be stored in the central database of Open eClass. This function is responsible for the storage of the title, the URL and the description that characterize a learning object, as well as the category into which the learning object is classified. The major contribution to the storage process is that we abolish the limitation of the storage of one link each time. The modified function supports a batch storage process of many links at once. In addition, this back-end modification makes feasible the automatic generation and storage of new categories for links classification. Until now, the insertion of links and categories were two discrete actions for the instructor. Before storing the learning objects metadata and categories, a check in order to avoid double-entries takes place, i.e. the same learning object of the same category will not be stored more than once. This operation requires communication with the central database of the Open eClass (MySQL, PHP).

– *Communication with external web services*: The *Link* module is modified appropriately to retrieve learning object metadata in XML format, either from XML sources or remote web services or from online repositories directly. In the first case, a simple function of XML file parsing has been developed (PHP). Each collection should always have a predefined, unique metadata schema (see Section 3.3). In the case of online retrieval, the module is capable of sending HTTP requests to external web services based on information that has been recovered as described in the function *Metadata retrieval for a course*, i.e. using course keywords as query terms. The external service to be queried can be configured within the main system configuration. The responses (XML and JSON format) are sorted and can be transformed in order to conform to the uniform metadata schema. The communication with the web services via the HTTP protocol is implemented using AJAX technology.

– *Logging instructors' actions*: Furthermore, we have implemented a logging process so as to gather data on i) how many learning objects are loaded from the external collections or how many results are retrieved for each keyword (especially when communicating with external services and/or LORs), ii) where the learning objects are coming from, iii) which instructor submits the load/search request (in the case of retrieval from external services we record also the specific set of keywords), iv) the date-time of each load/search request and v) how many of the results the instructor selects to save as links in his own course. We have adopted an external NoSQL-document database for the storage of all this information that is also configurable.

3.3 Alignment Procedure

When ingesting learning object metadata into the Open eClass platform, it is necessary to express them in compliance with the rules of the metadata schema that we propose in the context of this work. This schema consists of the five major elements as they are presented in the following table.

Table 5. Example instantiation of the metadata schema

```
<Results>
    <Category keyword="medicine">
        <Result>
            <URL>http://dx.doi.org/10.1097/JSM.0000000000000175</URL>
            <Title>American Medical Society for Sports Medicine position statement: interventional
                musculoskeletal ultrasound in sports medicine.</Title>
            <Description> (PUBMED) The use of diagnostic and interventional ultrasound has significantly increased over the past decade. A majority of the increased utilization is by nonradiologists. In sports medicine, ultrasound is often used to guide interventions such as aspirations, diagnostic or therapeutic injections, tenotomies, releases, and hydrodissections.
            </Description>
        </Result>
    </Category>
<Results>
```

As an initial step of the ingestion process, we need to configure not only which LORs we intend to utilize for the learning objects retrieval, but also how the responses will be efficiently transformed so as to manipulate them in one unified manner. It is necessary to create mapping rules in order to match the incoming results to the metadata of links (URL, URL title and URL description) according to the metadata schema. All the setting properties can be defined in a configuration file having JSON format. The following table shows through an example all the needed attributes that must be set so that the retrieved learning objects metadata from a new source would be consistent with the unified metadata schema.

Table 6. An excerpt of the configuration file

```
{
    "repositoryA": {
        "repository_url":"http://www.ebi.ac.uk/europepmc/webservices/rest/search/query=",
        "repository_extra_parameter":"&resultType=core",
        "metadata": {
            "result": "result",
```

```
            "url": "url",
            "url_title": "title",
            "url_description": "abstractText"
        }
  },
      "repositoryB": {
      "repository_url":"http://openarchives.gr/opensearch/",
      "repository_extra_parameter":"/limit:25",
      "metadata": {
          "result": "entry",
          "url": "content",
          "url_title": "title",
         "url_description": "dc:identifier"
      }
  }
}
```

In brief, the attributes `repository_url` and `repository_extra_parameter` are used in order to construct the HTTP GET requests. A typical URL in order to query the keyword ‘semantic’ in OpenSearch API is http://openarchives.gr/opensearch/semantic/page:1/limit:25 and consists of the following parameters:

– http://openarchives.gr/opensearch/: the main url of the script that responds to HTTP GET request
– *query*: keyword with which you want to search, in our example is ‘semantic’
– *page*: page number from which results start
– *limit*: the number of results per page

In our application, the parameter *query* should not be defined in the configuration file, since it is set ad lib from the end user at the front end. All the other attributes under the *metadata* attribute define how all the various responses can be translated in a uniform xml. The *result* attribute indicates how we can identify each result node in the xml response. Next, the *url*, *url_title* and *url_description* parameters allows us to distinguish the three elements from the rest of metadata that describe the result.

4 A Usage Scenario

After successful authorization, the logged instructor can select the *Link* module from the navigation menu of his course and then the *Add Learning Objects*. When an external harvesting service is to be used, a search form appears with a unique field that has a predefined set of keywords (see Figure 3).

These keywords, separated by a comma, include the keywords that the instructor has already set for his own course (section 3.2). However, the instructor is free to set a different set of keywords each time. The terms must always be separated by a comma so that they can be handled as distinct keywords by the recipient web service. In case the system is fed with a precompiled collection of metadata the keywords are simply ignored.

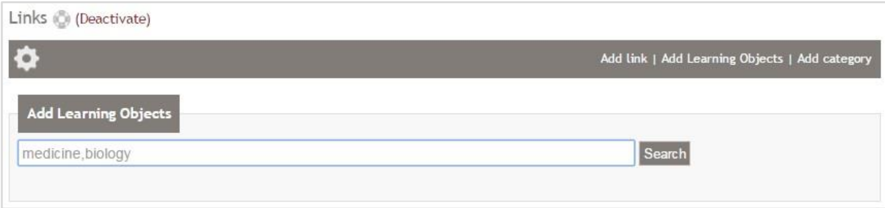

Fig. 3. Communicating search keywords to the recipient web service

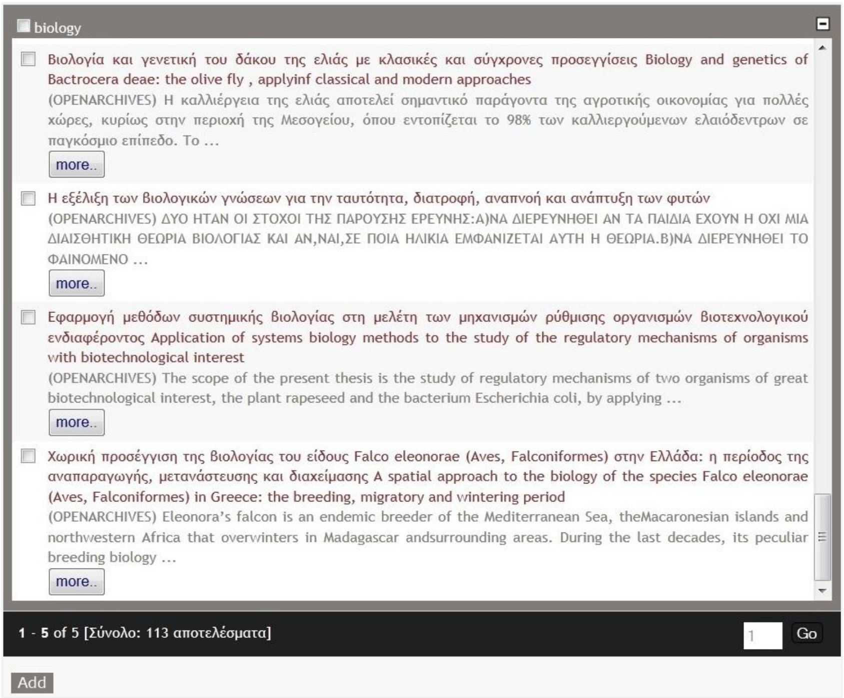

Fig. 4. List of learning objects – Front end

Once our application completes the loading process, the retrieved results, categorized based on each keyword, are presented in the form of a table (see Figure 4), under the search form. For clarity reasons, there is a pagination capability of the results' categories. Besides, the categories are shown by default collapsed and not the entire list of results is presented at once.

The instructor can traverse through the results' pages using the navigation buttons. The full list view of results for a particular category/keyword can be toggled by clicking on the plus/minus button or on the category title, which expands or collapses category results, respectively. Moreover, in order to pick a link for insertion, he can select the checkbox near the link title. In case he desires to select all the results for a category, he can do it at once by clicking the checkbox near the category title. Whenever he is ready to submit the selected links, he just pushes the button *'Add'*.

5 Conclusions and Future Work

Achieving interoperability between repositories of learning resources and LMSs is a key challenge and critical for the efficient dissemination, sharing and reuse of the huge amount of knowledge they manage. In this paper we presented a method for bridging an LMS with external learning object collections. Going beyond batch importing, we have built on top of a modular architecture and we have additionally designed and implemented a workflow for presentation, traversal and selection of external resources to be followed by instructors.

While different metadata schemata may pose barriers for direct integration, it is possible to identify a least common set of elements by referencing well known educational metadata standards such as LOM. We have shown that this set can form the basis for a common schema to be used for immediate ingestion of learning objects into eClass.

Although we specify a procedure for aligning LORs' schemata with our own, the ingestion process can be fully automated by delegating this task to a specialized web service. In this sense, the eClass service we have developed is extensible, in that it is already capable of communicating with external services through requests and responses. Future work would consider the development of such a service, which would perform federated search across repositories and then feed the metadata ingestion subsystem of the LMS, thus aiming at the development of improved knowledge retrieval services that make educational material search and reuse more efficient and straightforward.

Acknowledgements. This work has been partially supported by the project "Information System Development for Library Functional Services" of the Democritus University of Thrace, co-financed by Greece and the European Union, in the context of Operational Programme "Digital Convergence" of the National Strategic Reference Framework (NSRF) 2007-2013.

6 References

1. Gallagher, P. S. (2007). Assessing SCORM 2004 for its Affordances in Facilitating a Simulation as a Pedagogical Model. Doctoral Dissertation, George Mason University, Fairfax, VA.
2. Garcia, R., Pariente, T. (2009). Interoperability of Learning Objects Copyright in the LUISA Semantic Learning Management System. Journal Information Systems Management. Volume 26, Issue 3, pp. 252-261.
3. Gavidia, A.R., Sicilia, M.A., Garcia-Barriocanal, E., Palazuelos, G.M. (2008). Towards automated Specifications of Scenarios in Enhanced learning Technology, International Journal of Web-Based Learning and Teaching Technologies, Volume 3, Issue 1, pp. 68-77.
4. GUnet asynchronous eLearning group. Platform description (Open eClass 2.10). Retrieved from: http://docs.openeclass.org/en:detail_descr
5. GUnet Open eClass Asynchronous eLearning Platform (Teacher Manual). Retrieved from: http://www.ekpa-fa.gr/openeclass/manuals/mant/OpeneClass24_ManT_en.pdf

6. IEEE LTSC (2002). Draft Standard for Learning Object Metadata (IEEE 1484.12.1-2002). Available from: http://grouper.ieee.org/groups/ltsc/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf
7. IMS (2004). Metadata Best Practice Guide for IEEE 1484.12.1-2002 Standard for Learning Object Metadata. Version 1.3 Public Draft.
8. Karampiperis, P., Kastradas, K., & Sampson, D. (2003). A schema-mapping algorithm for educational metadata interoperability, Edmedia-World Conference on Educational Multimedia, Hypermedia & Telecommunications.
9. Lagoze, C., Van de Sompel, H., Nelson, M., & Warner, S. (2002). The Open Archive Initiative Protocol for Metadata Harvesting. Retrieved from: http://www.openarchives.org/OAI/openarchivesprotocol.html
10. Leal, J. P., & Queirós, R. (2010). Integration of repositories in Moodle, 8th National Conference on XML, Associated Technologies and Applications (XATA 2010), pp. 57-68.
11. M. Eap, M. Hatala, D. Gasevic (2008). Technologies for enabling the sharing of Learning Objects, International Journal of Advanced Media and Communication, Volume 2, Number 1.
12. MERLOT (2012). Multimedia educational resource for learning and on-line teaching website [Online]. Available at: http://www.merlot.org/
13. Namuth, D., Fritz, S., King, J., & Boren, A. (2005). Principles of sustainable learning object libraries. Interdisciplinary Journal of Knowledge and Learning Objects, Volume 1, pp. 181-196.
14. Ochoa, X., Cardinaels, K, Meire, M. and Duval, E. (2005). Frameworks for the Automatic Indexation of Learning Management Systems Content into Learning Object Repositories, Edmedia-World Conference on Educational Multimedia, Hypermedia & Telecommunications.
15. Stratakis, M., Christophides, V., Keenoy, K. and Magkanaraki, A. (2003) E-Learning Standards - SeLeNe (Self E-Learning Networks IST-2001-39045), Project Deliverable 2.1, Greece.
16. Szabo, M. and K. Flesher (2002). CMI Theory and Practice: Historical Roots of Learning Management Systems. World Conference on E-Learning in Corporate, Government, Healthcare and Higher Education, Montreal, Canada, AACE.