

How Content Management Problem of a Remote Laboratory System can be handled by integrating an open source learning management system? Problems and solutions

Burcu ÖZDOĞRU
Bilkent University
Ankara, Turkey

Nergiz Ercil CAGILTAY
Atılım University
Ankara, Turkey

ABSTRACT

This paper represents the design and implementation of the integration process of an open source learning management system (LMS) to the remote laboratory platform. The reason of using a learning management system is to prevent the problems which can be seen in the learner side of the remote laboratory systems. However, since using a learning management system such as Moodle can handle the learner problems, since it is a separate system, there are still other problems to be handled such as integrating it with the other parts of remote laboratory systems. This study explores the background information of LMS, the problems faced while integrating Moodle to remote laboratory applications, the chosen content management systems' architecture, our architecture for integrating Moodle with the remote laboratory system and the solutions we propose for the problems.

Keywords: Learning Management Systems, Moodle, remote laboratory systems, scorm

I. BACKGROUND INFORMATION

While developing remote laboratories, it's a must to take necessary precautions of the devices used in the experiments. The forthcoming damages to the devices cause loss of money, time and also decrease in learning performance. It is possible to prevent these forthcoming damages by developing a platform inside the remote laboratory system, which supports learners by providing information about the devices and experiments as well as assesses the performance of the learners to better facilitate their learning. Such remote laboratory systems support learners anywhere and anytime by providing information and practice abilities via the World Wide Web. The intension is to ensure the users or instructors in different countries to communicate with each other and share their experiences about the given devices or experiments.

While dealing with such systems, it is important to increase students' efficiency in the courses.

Recent studies done in learning environments [1] shows us reading from a material decrease the students' performance. For increasing motivation most of the institutions prefer to develop web pages including announcements, course contents, lab works (if any) and exercises. By this way students can reach the related content in any where at any time. It is possible and easy to create a web page for course instructors and nowadays most of the instructors or assistants have their own course web pages for students to follow up the announcements, coming events, contents, assignments and exercises. On the other hand, for some courses these items are not enough for the instructors. They want to publish interactive exercises maybe exams which aim students to enter values and get results. This could be done in web pages however, according to [1] there are some problems such as the system do not offer efficient tools to evaluate individual work of each student, there is no capability of a log system and course publishers must have a backup system and do the upgrades which could be annoying for some time passed.

What we need is to develop a platform to keep the students, logs, backups also the courses, contents, assignments, exercises, exams, grades etc. On the other hand our main aim is to develop this system for the usage of a remote laboratory system so, it is important to build a strong environment which has the ability to handle and manage the learner processes of the system in any where at any time.

This approach is known as LMS in the field of education and the goal is to provide standards for describing learning resources, communication protocols between learning resources, systems to manage the overarching delivery and handling of learning resources. In this approach, researchers and developers are having an agreement on handling the learning objects (LO), which are the piece of information, by a learning management system [2]. In short, LMS is a software application or web based technology used to plan, implement and assess a specific learning process and allow learners to authenticate themselves, register for courses, complete courses and take grades. Typically an LMS provides an instructor

with a way to create and deliver content, monitor student participation and assess student performance. It may also provide students with the ability to use interactive features such as threaded discussions, chat, mentoring, certification training, video conferencing and discussion forums. According to these needs, the Advanced Distributed Learning group has created a set of specifications called Sharable Content Reference Model (Scorm) to encourage the standardization of LMS.

Developing a learning management system which also supports international standards in the field, is not an easy task and requires high costs, since the analysis, design and the different implementation needs such as the user interfaces, database design and the communication requirements. Also, [4] states that such systems must have secure access to all tools and resources, and easy to integrate with other systems. An alternative solution is to choose a learning content management system among the existing ones. The important issue is to find a user friendly environment which our users can easily adopt themselves in the platform and start using it very quickly. Since our users can be from different countries it is a must to provide a multilingual environment. Another issue is related for the developers of the system. Our LMS system is a part of another system so it is obvious that in some time we need integration between these systems and with commercial LMSs it is not possible to change the code whenever we want. However, with an open source LMS we become independent and by some changes we can adopt the learner side to the other parts of the main system.

According to our research [3] [4] [5], Moodle [10] is the best solution for our content management problems since it contains both the learner management and content management features under the social constructivism approach. Nevertheless, Moodle's open source architecture gives us the ability of changing the structure according to our needs.

II. PROBLEMS WHILE INTEGRATING MOODLE TO REMOTE LABORATORY APPLICATIONS

Although Moodle's existing structure gains us time and money, it is not enough to solve our problems about the learner side of remote laboratory systems. The first problem to be solved is the Moodle's language options. Moodle offer language selections to the users however; this

option is limited only with the existing modules not the embedded content. Remote laboratory systems' motive is to reach everyone dealing with the necessary content and experiments therefore; we can not expect our users to use only one specific language such as English. [3] states that it is possible to create multilingual content for resources edited within Moodle such as text or html but it is still an issue to be solved. The other important problem is related with the Moodle's content management system. The modules and interfaces are all arranged in an educational manner therefore; the test system and the experiment platforms of remote laboratory systems are not available in Moodle's interface. The importance of knowing the necessary content about the devices or topics of the lessons have the higher priority to prevent the forthcoming damages to the devices so; we need to integrate the content stored in the Moodle with the experiment user interfaces. Otherwise it would be hard for the learners to switch between different systems and search for related information. This is one of the challenging issues that should be solved in order to integrate Moodle to the remote laboratory system. Since all platforms are bound to each other the necessary implementations should be done by the help of Moodle's open source architecture.

III. STRUCTURE OF MOODLE

For dealing with these problems it is important to understand the structure of Moodle. For the learner side of remote laboratories we are using two types of users generally. These are teachers and learners. Basically teachers are putting the content inside a course and learners will join to that course and examine the content. In Moodle, it is an easy task for teachers to develop and put the content however the main discussion is, in which format the teacher should develop the content. It is possible to develop content as a word document, html document or a zip folder. Although all of these has their own advantages the general idea is to develop content as a Scorm object in order to provide learner standards. In the early versions of the Moodle, developing the content as a Scorm object is doable however because of having some problems, the Moodle community strength this issue in the newer versions.

Scorm uses content packaging which describes data structures that are used to provide interoperability in order to define a standardized set of structures that can be used to exchange content. Since it is a need to exchange collections

of digital resources between different learning management system's, authoring tools, content repositories and operating systems, using a content package is a necessity. In traditional instructional design terms, the content package would be everything needed to deliver to the learner such as a course, module or lesson which is important for us since we want to use our course as a Scorm object in Moodle.

Since, it is obvious that Scorm used in educational society for increase sharing among different users, creating courses as a Scorm object in Moodle make sense. Another reason is about our goals; we want to collect resources about the devices learnt in a course, combine them and change them whenever we want with small interventions. If we think of these combinations as objects containing lots of information inside, it is an easy task for us to manipulate them. Also while doing this we want to provide sharing and standardization. From this point of view what we need is learning objects (LO). Learning object can contain any number of different items such as a web page, a picture, a flash animation, in short any element inside a course [6]. The important point to be considered in learning objects is that; every learning object has a common language to communicate. By this way, using a learning object or adapt it to another system causing no problems for developers. This common language is called XML; which is stored in plain text format. XML has become both machine-usable and human-readable, providing software and hardware independent way of storing data. This makes it much easier to create data that different applications can work with. It also makes it easier to expand and upgrade a system to new operating systems, servers, applications and new browsers. Since XML is independent of software, hardware and application, data are available to other than only standard browsers [7].

IV. CONTENT PACKAGES FOR USING AS SCORM OBJECTS

Moodle offers only uploading prepared Scorm objects inside a course, meaning instructors or developers cannot prepare a course as a Scorm object inside Moodle. This problem diverts us to search for a tool to create content packages for using as Scorm objects. Since there are several tools available, we prefer to use a tool offered by Reload project which simply is a Content Package and Metadata Editor. The aim is to take the content users want to use, package and describe it [8].

According to our research, creating packages is easy with Reload tool however; another important issue must be taking under consideration, which is metadata. Briefly, metadata is data about data and is the common property of all learning standards and specifications. It became important on the World Wide Web because of the need to find useful information from the mess of information available. There is a relationship between metadata and the information resource it describes. This means that, there is some information describing a resource which, we call metadata.

Scorm uses the metadata part of content packaging to track and locate materials. Here, the learning management system uses metadata to give information about content aggregation (course, module, and lesson). When learning resources are created, ideally the author also creates the metadata that describes the learning resource so that it can be located and reused elsewhere. Such metadata is considered context independent of a particular collection that comprises a specific learning strategy.

Reload tool [11] gives us the LOM information model and the creators can easily adopt the related metadata of the course inside a content package. Since the contents in our system are collected from different users, creating a content package gives us independency. However; it is important to get necessary metadata from the content creators for further usage. In accordance with our goals, it must not be forgotten that Moodle is only a part of the general system named as ERRL. In some point we need integration with the other parts of the system and eventually we need to use the metadata we collect from the content creators. For the beginning the necessary metadata information needed are;
general.title, general.language, general.keyword,
general.description, lifecycle.version,
lifecycle.status, technical.format,
technical.duration, technical.size,
technical.location, technical.requirement,
technical.requirement.type,
technical.requirement.name,
educational.difficulty,
educational.typicalLearningTime and
annotation.date.

It is obvious that, content creators can also create content packages and metadata of the related content however; this opinion causes some problems such as, all the content creators should learn reload tool and the usage of metadata which will be a long process. In the long term, this

option causes confliction between different content creators so; the reasonable method is to set a team responsible for collecting the resources and metadata from the creators and then create the content package and upload it to the Moodle.

The first thing to do after all these researches done in this area is to create a content package by using reload tool and upload it inside Moodle. The draft content is about Vector Network Analyzer and we have only one HTML page.

If the information is valid, the content package can be developed by using the zip option. The next step is to put this zip folder inside the Moodle which is an easy task to achieve. After uploading the zip folder, the content appears as an iframe.

V. OUR ARCHITECTURE FOR INTEGRATING MOODLE WITH THE REMOTE LABORATORY APPLICATIONS

Since we have completed a big step in our system as deciding to upload what type of content inside Moodle and achieve the processes successfully, we have to deal with the other problems such as search options and integration of the search mechanism of the ERRL system. The system we try to achieve can be examined in Fig 1.

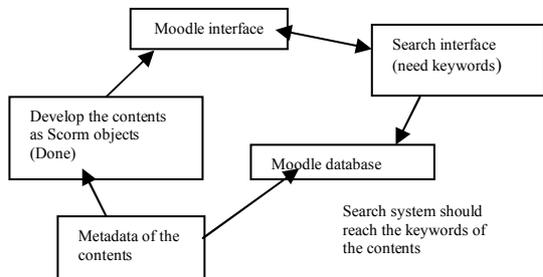


Figure 1 – The architecture of the system

The search system’s aim is to give students or learners information about the content they seek for. The system is independent from Moodle interface like the experiment interface however; on the contrary of the experiment interface, search system can be reached inside Moodle simply by using a web link. According to the system we try to build, what we need is to reach the related information defined in the content’s metadata. So far we developed two suggestions in theory.

The first suggestion is to use directly the Scorm objects’ defined metadata information. Before uploading Scorm objects in Moodle we have to create the content package of the resources of the

related course, and while developing the content package we define the metadata taken from the content creators meaning the instructors. After developing the content package, Reload editor produce an XML file named as imsmanifest.xml, which contains both the information of the resources inside the package and the metadata of the package which the content creators gave.

Since there are two different parts of creating and uploading a content package, it is important to understand both parts and the relationship between them. In the Moodle side, when the users uploading a Scorm object, only the prepared content and resources appeared in Moodle’s interface, it is not possible to see the XML file or schemas of the package. In order to understand the architecture of an uploaded Scorm object, Moodle’s database should be examined.

In the database, the related tables which we can use;

- a) mdl_scorm; which represents each Scorm module and its configuration.
- b) mdl_scorm_scoes; which represents each SCO part of the Scorm module.

However; both tables do not show the XML file or the metadata of the content package. It is obvious that, there must be a relationship between these tables and XML file. According to the comparison done between imsmanifest.xml file and mdl_scorm_scoes table’s values, it appears that both the value of manifest column of mdl_scorm_scoes table and the identifier value of imsmanifest.xml file are the same; which can be seen in Fig. 2 and Fig. 3.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE scorm SYSTEM "http://www.imsglobal.org/xsd/imsdcp_v1p3.xsd" [
  <manifest xmlns="http://www.imsglobal.org/xsd/imsdcp_v1p3.xsd"
    xmlns:imsmd="http://www.imsglobal.org/xsd/imsdcp_v1p3.xsd"
    xmlns:adlcp="http://www.adlnet.org/xsd/adlcp_v1p3.xsd"
    xmlns:imsseq="http://www.imsglobal.org/xsd/imsseq_v1p3.xsd"
    xmlns:adlnav="http://www.adlnet.org/xsd/adlnav_v1p3.xsd"
    xmlns:scorm="http://www.imsglobal.org/xsd/imsdcp_v1p3.xsd"
    xmlns:scorm="http://www.adlnet.org/xsd/adlcp_v1p3.xsd"
    http://www.imsglobal.org/xsd/imsseq_v1p3.xsd
    http://www.adlnet.org/xsd/adlnav_v1p3.xsd
  ]>
  <metadata>
    <schema>ADL SCORM</schema>
    <schemaversion>2004 3rd Edition</schemaversion>
    <general>
      <title>
        <imsmd:string>Network Vector Analyzer</imsmd:string>
      </title>
      <language>
        <imsmd:language>en</imsmd:language>
      </language>
      <description>
        <imsmd:string>The draft content of VNA</imsmd:string>
      </description>
    </general>
  </metadata>
</manifest>
```

Figure 2 – imsmanifest.xml file of the content package

id	scorm	manifest	organization	parent
92	7	MANIFEST-ADD97DBC46919102C89216FB76C73080	ORG-ARE8AAE61EE0F1968F4059E10BACD0699	ORG-ARE8AAE61EE0F1968F4059E10BACD0699
90	7	MANIFEST-ADD97DBC46919102C89216FB76C73080	/	/
91	7	MANIFEST-ADD97DBC46919102C89216FB76C73080	ORG-ARE8AAE61EE0F1968F4059E10BACD0699	ORG-ARE8AAE61EE0F1968F4059E10BACD0699

Figure 3 – mdl_scorm_scoes table’s values
From this point of view, it seems that by using the manifest column in the table we can reach the

metadata inside the `imsmanifest.xml`. By this way, reaching the metadata during the search process can be done automatically and it will be fast and efficient. However; there is still a question exists of how to reach the xml file and take the metadata inside it. This problem can be solved by developing another interface for linking the xml file and the table. This interface should query the learning objects from our learning management system to get the objects' metadata [9]. Developing an interface for this purpose could ease to combine the XML file and the table however; it could also cause other problems such as increasing the cost and time.

Our second approach for the search system is to create a table inside Moodle containing the metadata of the packages. Since we have the Scorm tables holding the resources inside the packages one by one, it should be easy to reach them if we define the Scorm objects' id number inside the new table. We can develop our new table; assume its name as `mdl_content_metadata`; containing the Scorm object's id, a representing metadata id and the metadata information such as the keyword, title, description etc. By simple queries, first the program reaches the `mdl_content_metadata` table and then the `mdl_scorm_scoes` table.

As an example, let's assume the user searches for the keyword VNA. When the search program processes first it goes to the `mdl_content_metadata` table and find the related keywords, from this table by the help of `scorm_id` column it reach the `mdl_scorm_scoes` table and find the related content and retrieve the found content back to the search system.

This theory is much easier to develop then the first one however; entering metadata twice, both in the reload tool and in Moodle's database may cause a problem in the long run. As an example, if there is a need occurs for a new metadata entry inside a table as a column, all of the past information should be updated by hand. For the first theory, this could be solved by changing the content package's metadata which can be reached from the XML file, since content packages have the property of reusability.

VI. CONCLUSION

To conclude; developing content packages and uploading them as Scorm objects in Moodle give us the ability of interoperability, reusability, accessibility and durability, and in our system we finished this part. However; the other issue of

reaching the metadata from the search interface is still a problem. In learning management systems, search module is a must to be used and Moodle also has this module. The problem here is we do not want to use Moodle's search module because as mentioned before, Moodle is only a part of the ERRL system used for the learning side. In ERRL; learner, search and experiment interfaces are not related to each other so; there must be an integration between them. For the integration of search and Moodle interfaces both of the theories mentioned above will be tested and according to their advantages and disadvantages, one of them will be used in the system.

VII. REFERENCES

- [1] F.Teixeira, M. Barata , M. Viera and H. Silva, "Migrating From a Web Site to a Moodle Based LMS", 2006. Retrieved April 06, 2007 from <http://www.euitt.upm.es/taee06/papers/S4/pd7.pdf>
- [2] G. Vossen and P. Jaeschke, "Learning Objects as a Uniform Foundation for E-Learning Platforms", *Seventh International Database Engineering and Applications Symposium (IDEAS'03)* p. 278, 2003.
- [3] Edutech, University of Fribourg, retrieved March 17,2007 from <http://www.edutech.ch/lms/ev3/showreport.php?prodid=2>
- [4] B. McMullin and , M. Munro, "Moodle at DCU", 2004. Retrieved March 17, 2007 from <http://odtl.dcu.ie/up/2004/odtl-2004-01.html>
- [5] B. Bos, K.D. Munoz and J.V. Duzer, "Blackboard vs. Moodle, A Comparison of Satisfaction with Online Teaching and Learning Tools", 2005. Retrieved November 01, 2006 from <http://www.humboldt.edu/~jdv1/moodle/all.htm>
- [6] S. Downes, "Learning Objects. U. of Alberta", 2000. Retrieved April 10, 2007 from <http://www.downes.ca/files/Learning-Objects.htm>
- [7] N. Walsh, "A Technical Introduction to XML" 2006. Retrieved January 5, 2006 from <http://www.xml.com>
- [8] Reload, "Reload Editor Introductory Manual", 2004. Retrieved May 15, 2007 from <http://reload.ac.uk>
- [9] J. Broisin, P. Vidal, M. Meire and E. Duval, "Bridging the Gap Between Learning Management Systems and Learning Object Repositories: Exploiting Learning Content Information", 2005. *Proceedings of the Advanced Industrial Conference on Telecommunications/Service Assurance with Partial & Intermittent Resources Conference/E-Learning on Telecommunications Workshop.*
- [10] Moodle, <http://moodle.org>
- [11] Reload tools, <http://www.reload.ac.uk>