Journal of Universal Computer Science, vol. 13, no. 7 (2007), 924-931
submitted: 7/3/07, accepted: 25/7/07, appeared: 28/7/07 © *J.UCS*

A First Step Mapping IMS Learning Design and Moodle

Daniel Burgos
(Open University of the Netherlands, The Netherlands
daniel.burgos@ou.nl)

Colin Tattersall
(Open University of the Netherlands, The Netherlands
colin.tattersall@ou.nl)

Martin Dougiamas
(Moodle, Australia
martin@moodle.org)

Hubert Vogten
(Open University of the Netherlands, The Netherlands
hubert.vogten@ou.nl)

Rob Koper
(Open University of the Netherlands, The Netherlands
rob.koper@ou.nl)

Abstract: Mapping the specification IMS Learning Design and the Course Management System Moodle is a logical step forward on interoperability between eLearning systems and specifications in order to increase the best acceptance of the specifications into the widespread world of the eLearning systems and to ensure the standardization of the outputs from the systems to be used in others. IMS Learning Design and Moodle look for a common understanding focused on the integration of information packages modelled by each part in the other. The final goal aims at Moodle playing an IMS LD package. A second step will map a Moodle course to be used in any IMS LD complaint tool. The Unit of Learning in IMS LD and the course in Moodle become the perfect couple where to find several elements that should match each other. This paper shows how to make this understanding, mapping related elements in both to get a list of pairs easy to translate from one to another, and to define also a list of requirements for this protocol.

Keywords: IMS Learning Design, Moodle, Integration, Mapping
Categories: H.4.2, D.2.1, D.2.12

1 Introduction to IMS Learning Design and Moodle

The mutual understanding between IMS-LD and Moodle seems like an improvement for both parts. IMS-LD provides a pedagogical flexible approach in the creation of UoLs, as well as the specification support and the technical background focused on standardization and interoperability; and Moodle provides a well-known and easy-to-learn CMS and an active community of non-technical profile. To this end, Moodle and The Open University of The Netherlands established a working group in June

2005, then hosted by the UNFOLD Project and Learning Network for Learning Design and currently supported by the ProLearn and TENCompetence projects.

1.1 IMS Learning Design

IMS Learning Design, or simply IMS-LD, [IMSLD 2003] is a specification to represent and encode learning structures and methods for learners and teachers. Furthermore, IMS-LD is focused on the design of pedagogical methods able to manage learning activities linked to learning objects within a learning flow. This learning flow consists of plays, acts, activities, activity structures and environments and it is flexible enough to provide several personalized itineraries depending on the role assigned or on a set of rules. The usual life-cycle starts with a lesson plan modelled according to the IMS-LD specification, defining roles, learning activities, services and several other elements, inside a XML document called Manifest [Tattersall et al. 2003]. An information package written in IMS Content Packaging [IMSCP 2001] is used as a container of the resources and links them with the IMS-LD structure, defined in the Organizations section of the IMS CP package. Later, the Manifest is packaged with the nested resources in a compressed ZIP file, meaning a Unit of Learnig (UoL) [Koper and Tattersall 2005]. Therefore, the UoL is a compressed file with a) an XML manifest, describing method, plays, acts, roles, activities, environments, properties, conditions and/or notifications of the Learning Design specification and also pointing to the related resources; and b) a set of files or resources mentioned in the XML manifest. Once the UoL is validated, published and run in a player, it will player will coordinate the teachers, the students, the activities during the learning process. A user will follow the learning path created for him or for his role and will carry out the related activities in order to complete satisfactorily the Unit of Learning.

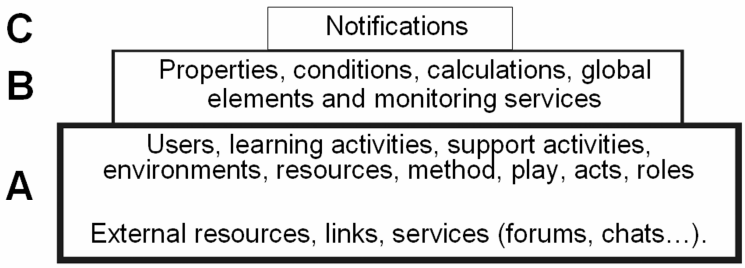

Figure 1: IMS Learning Design and three levels specification

IMS-LD consists of three levels (Figure 1). Each level itself provides specific features to the Unit of Learning. Level A provides method, plays, acts, roles, role-parts, learning activities, support activities and environments; Level B provides properties, conditions, calculations, monitoring services and global elements; and Level C adds notifications. Every layer is made upon the previous one. Level A is the main part of the specification, as it provides the baseline to build any Unit of

Learning, Level B adds powerful features to create more complex e-learning lesson plans, and Level C sums a specific and useful trigger element.

Every single step between the creation and the use needs an IMS-LD compliant tool. The UoLs can be created with general purpose editors or with specific IMS-LD editors, like CopperAuthor [Van der Vegt 2005], Cosmos [Miao 2005] or Reload LD Editor [Bolton 2004], and they can be run with several tools and engines, like CopperCore [Vogten and Martens 2005] or Sled [OUUK 2005]. However, current tools do not allow for an easy editing of rich and flexible UoLs. They make the creation of adaptive UoLs technically possible, but too difficult for a non-technical user. It is the same with general purpose editors, like XML Spy. A creator has to know the technical editors and the specification in depth. A higher level layer with a more visual metaphor is still missing, although on its way. The TENCompetence Project, for instance [TENCompetence 2005] is developing a visual LD Editor. Also, Complutense University is developing the <e-LD> project, using UML diagrams to represent the flow of the different Units of Learning. Nevertheless, for the time being, a significant effort is still needed to create adaptive IMS-LD UoLs. As an example, there is a repository with a number of runnable sample UoLs [LN4LD 2005; OUNL 2002] focused on several educational issues.

1.2 Moodle

Moodle [Dougiamas 2003] is a Course Management System (CMS) easy to install and to use and wide abroad disseminated, with more than 100.000 registered users, 12.000 registered websites and translation into 70 languages. Also, Moodle has a very strong virtual community of active users carrying out an increasing amount of face-to-face and online activities, and supporting each other via the official site and a number of ad hoc assorted websites. Moodle is able to manage every feature of a course and the related environment, such as user definition, groups, access, resources, internal links and a long et cetera. This is a difference with IMS-LD. From a social constructionist approach a course is created from scratch in a few minutes. Moodle also allows for the execution of other information packages, like SCORM, LAMS and Content Packaging, as an encapsulated module inside a course. On the other hand, the pedagogical expressiveness of Moodle is limited by the absence of features included in the IMS-LD model, such as defined adaptive learning flow, flexible roles and adaptive content. Currently, Moodle is working intensively to provide these missing features so that it is able to support IMS-LD more fully over time. The main issue is that IMS-LD and Moodle are not comparable at all, as they have different approaches to work on e-learning as well as they are different releases [Berggren et al. 2005].

2 Basic approach on mapping IMS-LD and Moodle

There are a few attempts to integrate UoLs with stand-alone LD players like Sled and Moodle [Burgos and Corbalan 2006]. Sled is able to run a UoL stored in a LD server via an Internet Explorer player in the client (Telcert). A link from a Moodle resource to the ip address where Sled is allocated allows a simple first level of integration. This structure is not focused on the reusability of the lesson plans but on the integration of current systems to form a de facto more complex approach, collecting several

technologies. Other attempt looks into the communication between the IMS-LD based engine called Coppercore and external systems and services, like <e-Adventure>, that executes adaptive educational games fully integrated in Units of Learning [Burgos et al. 2007]

What the mapping of Moodle and IMS-LD aims to is the re-usability of a lesson plan/course/UoL of Moodle into an IMS-LD compliant tool, or the other way around; this re-usable information package could be used as a base for a further development or as they are actually defined. Furthermore, this mapping is focused on the interoperability and the reusability of an information package/UoL, no matter the original platform that is used for it. Moodle has already started this mapping exercise with LAMS, SCORM and IMS Content Packaging.

In order to achieve the best understanding between IMS-LD and Moodle, we define a three step process:

1. Moodle is able to export one course to a UoL, translating the Moodle mark-up internal language to the IMS-LD XML based notation. In this first step, we find couples that identify specific elements of Moodle and their mirrors in IMS-LD. These elements represent the core of a Moodle course. Additional elements that are not required could be left aside of mapped as external services.
2. Moodle is able to import one UoL into the Content Management System and translate the IMS-LD notation into a Moodle notation. The biggest challenge is focused on the translation of Level B elements in Moodle, since Moodle does not support several of such elements, like conditions and global elements.
3. Moodle is able to play a UoL inside the system, following one of the approaches on types of integration between information packages and players suggested in [Tattersall et al. 2006]: Moodle stores an IMS-LD information package and it runs an internal player. This integration can be implemented in several ways: a) a full integration where there is a communication of properties and states between both parts; b) a half-way integration where the information package is inside the other system, but there is no communication between them; and c) a low integration where there is a link between both applications, but the information package is not hosted in the foreign system.

To realize these three steps we need to establish a general framework, with some restrictions. Since both parts are different in essence (Moodle is a CMS, while IMS-LD is a specification) a constraints list is required in order to find a common agreement which the understanding could be built upon. Specifically:

1. Besides the Moodle course, the rest of the Course Management System environment is out of scope (calendars, blocks, log-in, language…) as they are used as processes and instructions and not like a core part of the basic unit of interchange for information (i.e., a Moodle course or an IMS-LD UoL).
2. There is a need for matching every single Moodle feature-component to an equivalent in IMS-LD or to define it like an external process/instruction.
3. In order to make taxonomy of the elements in a Moodle course and to find a mirror in the IMS-LD specification, we define four main groups: 1) Setting (basic configuration), 2) Activity, 3) Resource, 4) Administration (out of scope).

If we take a Moodle course we can match every element of the course with an element in the basic structure of an IMS-LD Unit of Learning. Table 1 shows the result:

Issue	Moodle feature-component	IMS-LD structure	Remarks
0	Full course	1 UoL, 1 play, 1 act, 1 activity structure (type selection)	
1.1	Setting: Full name	Title of Learning Design	
1.2	Setting: Short name	Title of Play	
1.3	Setting: Hidden sections	Activity Structure:IsVisible	Possible, but suggested for a next iteration
1.4	Setting: summary	LD learning objectives	
1.5	Setting: Your word for Students	roles:learner:title	
1.6	Setting: Your word for Teachers	roles:staff:title	
2.0	Topic	Learning activity	If a resource or an activity is defined there will be an environment If there is no activity description in Topic it should be auto-generated with a standard text or just empty
2.1	Topic 0	Learning activity (first one)	
2.2	Summary of Topic	Activity Description	
2.3	Activity: Chat	Synchronous conference (conference-type)	Services
2.4	Activity: Forum	Asynchronous conference (conference-type)	Services
2.5	Activity: Glossary	-	Generate as dynamic HTML
2.6	Activity: Quiz	-	Possibly it generates QTI
2.7	Activity: Scorm	No match	For a next iteration
2.8	Activity: Survey	External	Generated by Level B
2.9	Activity: Workshop	External	Generated by Level B
3.0	Resource	Learning object	All the resources ended as learning objects except Directory and Link
3.1	Resource: Directory	-	Generated by Level B
3.2	Resource: Label	Learning object	Could map to a title of environment if needed
3.3	Resource: Link to file	Learning object	File should be included in content pack
3.4	Resource: Link to website	Learning object	Link should be an absolute URI

Table 1: Elements in an IMS-LD Unit of Learning mapped to elements in a Moodle course.

3 The first step: Exporting a Moodle course to an IMS-LD Unit of Learning

The first step in the integration process is focused on the exportation of a Moodle single course to an IMS-LD UoL. In order to achieve this goal we establish a list of assumptions: a) there is no round tripping for the first stage, b) the exportation is completely made in batch mode (therefore, no dialog nor user interaction), and c) this task is planned as an iterative process where the first iteration gets the basic skeleton for conversion and the subsequent versions will extend it.

IMS-LD is defined as a metaphor built around a theater using roles, plays and acts. Inside, some elements describe the educational framework: learning objectives, activities, environments, property of visibility, method, type of learning flow, etc. In Figure 1, there are several couples of elements IMS-LD & Moodle defined in the most basic structure of a Moodle course:

IMS Learning Design UoL	Moodle course	
Unit of learning	0	1 UoI, 1 play, 1 act, 1 AS (type selection)
Title	1.1	Full name
Learning objectives	1.4	Summary
Roles	1.5	Students
	1.6	Teachers
Play		
Title	1.2	Short name
Act		
Title		
Activity structure		
Title		
isVisible	1.3	Hidden sections
Learning Activity	2.0, 2.1	Learning Activity
item-description	2.2	Summary of topic
Learning Objectives		
Environment	Several	Activity and resource: External, Learning Object or Conference
isVisible	2.14	Activity: Assignment: visible/hidden
	3.6	Resource: visible/hidden

Figure 2: Basic match between an IMS-LD UoL and a Moodle course.

4 Mapping services

Some activities in Moodle need a special process, as they have some basic data used for the appropiate execution (i.e., forum, wiki, quiz…). Every activity or resource in Moodle needs to export some additional information that is not supported by IMS-LD, i.e., timing in Forums or grades in Workshops. If IMS-LD is not able to manage this information it will be lost and no later retrieval will be possible from IMS-LD to Moodle, in future versions, although no round tripping is assumed in the first approach.

A possible way is to associate a file with all the extra information related to an activity. For instance, a Moodle Forum is matched to an asynchronous conference-

type service in IMS-LD and there is a linked file with the information about starting time, ending time, or discussions policy. We could call the file `serviceparams.xml`. This file is a resource in the content package, type `servicecontent`, although other types are possible: `webcontent`, `imsldcontent` and `imsqti_item_xmlv2p0`. The field serviceparams.xml needs to be associated with the service `service-conference`, but this is not possible yet in the current IMS-LD 1.0´s information model. One approach is to associate an additional learning-object with a service. The final approach is as follows:

```
  <imsld:environment>
       identifier="env-Topic-0-News-Forum">
       <imsld:title> Moodle Summary Topic </imsld:title>
          <imsld:service          identifier="service-conference"
isvisible="true">
            <imsld:conference conference-type="synchronous">
              ...
          </imsld:conference>
       </imsld:service>
       <imsld:learning-object identifier="ForumParams">
           <imsld:item                          identifier="PARAMS"
identifierref="RefToParams"/>
       </imsld:learning-object>
  </imsld:environment>

  …

  <imscp:resource                        identifier="RefToParams"
type="servicecontent" href="params.xml">
       <imscp:file href="params.xml" />
  </imscp:resource>
```

Following this structure we could map any resource or activity without any lost of relevant information in Moodle. Although this approach of Moodle is miss-using the notion of a learning object, it could serve as a temporary solution until a modification of the XML Scheme in a new version of IMS-LD could happen.

5 Conclusions

IMS Learning Design and Moodle are two entities that work on the process to reach a kind of integration that allows for the exportation of a Moodle course to an IMS-LD format, the importation a UoL to a Moodle CMS, and the execution of an IMS-LD information package into a Moodle CMS. Following this three-step process a working group formed by Moodle and The Open University of The Netherlands looks for an understanding between both eLearning parts. The final objective is to achieve some level of interoperability and re-usability of online lesson plans and courses.

The first step to take is focused on the exportation issue and it aims to get a basic mapping between both, taking a simple Moodle course that could be extended with additional features in a second round. For this exportation, a challenge is to keep some information in services or activities that Moodle uses for their configuration and execution. An approach points to keep this information in a file stored inside the information package as a resource and to link it as a learning object to the called

service. This way, all the needed information is fully exported to an IMS-LD package but leaving it out of the main manifest, where it could not be properly handle, as there are no possible match yet. A second approach points to a modification of the XML Scheme in IMS-LD 1.0 where this situation could be manage directly.

References

[Berggren et al. 2005]Berggren, A., Burgos, D., Fontana, J.M., Hinkelman, D., Hung, V., Hursh, A. & G. Tielemans: "Practical and Pedagogical Issues for Teacher Adoption of IMS Learning Design Standards in Moodle LMS"; Journal of Interactive Media, Special issue on Learning Design (2005).

[Bolton 2004] Bolton: "Reload Project" (2004), Available at www.reload.ac.uk.

[Burgos and Corbalan 2006] Burgos, D., Corbalan, G.: "Modelado y uso de escenarios de aprendizaje en entornos b-learning desde la práctica educativa"; III Jornadas Campus Virtual, Madrid (2006).

[Burgos et al. 2007] Burgos, D., Moreno-Ger, P., Sierra, J. L., Fernández-Manjón, B., Koper, R.: "Authoring Game-Based Adaptive Units of Learning with IMS Learning Design and <e-Adventure>"; International Journal in Learning Technologies (2007).

[Dougiamas 2003] Dougiamas, M.: "Moodle" (2003), Available at http://moodle.org.

[IMSCP 2001] IMSCP: "IMS Content Packaging" (2001), Available at www.imsglobal.org.

[IMSLD 2003] IMSLD: "IMS Learning Design" (2003), Available at http://www.imsglobal.org/learningdesign/index.cfm.

[Koper and Tattersall 2005] Koper, R., Tattersall, C., eds., *Learning Design - A Handbook on Modelling and Delivering Networked Education and Training* (Heidelberg, 2005).

[LN4LD 2005] LN4LD: "Units of Learning developed by several authors at Learning Network for Learning Design of The Open University of The Netherlands" (2005), Available at http://moodle.learningnetworks.org/course/view.php?id=20 and http://dspace.ou.nl.

[Miao 2005] Miao, Y.: "Cosmos LD Editor" (2005), Available at.

[OUNL 2002] OUNL: "DSpace" (2002), Available at http://dspace.learningnetworks.org.

[OUUK 2005] OUUK: "Sled player" (2005), Available at http://sled.open.ac.uk.

[Tattersall et al. 2003] Tattersall, C., Manderveld, J., Hummel, H., Sloep, P., Koper, E. J. R.: "IMS Learning Design Frequently Asked Questions" (2003), Available at http://dspace.learningnetworks.org/handle/1820/116.

[TENCompetence 2005] TENCompetence: "TENCompetence Project" (2005), Available at www.tencompetence.org.

[Van der Vegt 2005] Van der Vegt, W.: "CopperAuthor" (2005), Available at www.copperauthor.org.

[Vogten and Martens 2005] Vogten, H., Martens, H.: "CopperCore 3.0" (2005), Available at www.coppercore.org.