

Practical pedagogical uses of IMS Learning Design's Level B

Citation for published version (APA):

Burgos, D., & Koper, R. (2005). Practical pedagogical uses of IMS Learning Design's Level B.

Document status and date:

Published: 27/10/2005

Document Version:

Peer reviewed version

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

<https://www.ou.nl/taverne-agreement>

Take down policy

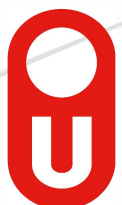
If you believe that this document breaches copyright please contact us at:

pure-support@ou.nl

providing details and we will investigate your claim.

Downloaded from <https://research.ou.nl/> on date: 02 Dec. 2023

Open Universiteit
www.ou.nl



Practical pedagogical uses of IMS Learning Design's Level B

Daniel Burgos and Rob Koper
Educational Technology Expertise Centre (OTEC)
Open University of the Netherlands
Valkenburgerweg 177; PO Box 2960
6401 DL Heerlen; The Netherlands
{daniel.burgos;rob.koper}@ou.nl

Abstract

One of the main concerns while making lesson plans in IMS Learning Design is how to model practical pedagogical actual scenarios in IMS Learning Design, and how IMS Learning Design can help to move real lesson plans, fully focused on pedagogical and didactical uses, to an open e-learning specification without getting lost in the process within technical issues. So, is it possible to make it? and how?

This paper intends to put together the pedagogical requests of teachers and learning designers and the technical approach needed to come them true using the Level B of IMS Learning Design. Through different examples and specific uses we describe both, the pedagogical needs and the suggested coding and we link them to provide a joint together view that allows to point out a discussion formula where didactical end-users needs on teaching meet a pedagogically expressive specification able to come across.

Keywords

IMS Learning Design, Unit of Learning, Level B, Adaptive Learning, Collaborative Learning, Assessment, Instructional Design

1. Introduction

IMS Learning Design, or just IMS LD (IMS, 2003), is a pretty new specification that jumped into the e-learning panorama to build a bridge from pedagogical face-to-face actual models to e-learning and blended learning frameworks. IMS LD allows to model regular lesson plans of teachers in units of learning to be run in a certain online platform. Since one year ago until now a number of tools (engines, players and editors) have been born, like CopperCore (Vogten and Martens, 2005), CopperAuthor (Van der Vegt, 2005), Reload (Bolton, 2004), Sled (OUUK, 2005) or several others and improved in parallel with some efforts on dissemination, like the European UNFOLD Project (UNFOLD, 2004) and Learning Network for Learning Design (OUNL, 2004). In addition, a few efforts have been made on interoperability between open source tools based on IMS LD or related to it, like Moodle (Dougiamas, 2004).

The Open University of The Netherlands (OUNL) works intensively providing engines, tools, examples and extensive documentation about the specification, the related ones, and feasible applications of them. In 2004 and 2005, OUNL also carries out the task of disseminating IMS LD, funded by UNFOLD, together with several European universities.

In 2005 a number of face-to-face meetings and some online activities haven been carried out by these two institutions and some strong useful feedback have been got from end-users, mainly from teachers and learning designers. One of the main concerns is how to move from an actual lesson plan in a classroom to a well structured Unit of Learning in IMS LD without losing pedagogical expressiveness. The second concern is how to create these units in an easy way for non-technical end-users.

2. Basic structure of IMS Learning Design

IMS Learning Design is divided into three levels. Level A, with the definition of the method, plays, acts, roles, role-parts, learning activities, support activities and environments. It is the core of the specification, contains the description of the elements that configure IMS LD and the coordination between them. For instance, role-parts define what activities must be taken by a role in order to complete an act and, subsequently, a play.

Level B, adds properties, conditions, monitoring services and global elements to Level A, and provides specific means to create more complex structures and learning experiences. Properties can be used as variables, local or global ones, storing and retrieving information for a single user, a group or even for all the characters involved. Through these mechanisms the learning flow can be changed at the run time, as decisions can be made taking into account dynamic content. Logically it is the used level to express the most of the pedagogical needs concerning adaptation, personalization, feedback, tracking and several other usual requests of teachers and learning designers.

Lastly, Level C adds notifications to Level B, meaning an email sent and a show/hide command to a specific activity, depending on the completion of another one.

3. Practical pedagogical uses

Level B provides several facilities concerning properties, conditions and the monitoring service, as we have already explained above, and all of them can be used in a wide range of applications. Following, the most often uses are described from a didactical perspective: active learning, collaborative learning, adaptive learning, personalization, dynamic feedback, runtime tracking, ePortfolios and new ways of assessment. Although the source code provided shows properly the dataflow and the information structure it is not the original code in IMD LD notation. Some superfluous details and additional tags have been ruled out in order to get a neater explanation. Besides, the engine CopperCore (*op. cit.*) has been used to run and try all the examples, representing the only current tool capable of a successful running of all the levels in the Specification.

3.1 Active and collaborative learning

Collaborative learning means (Cole and Engestrom, 1993) to share information in a peer basis, student-student and student-teacher, chasing to consolidate knowledge, criticize opinions and remarks, provide some new ideas in the light of others' work or insert new topics for discussion and a collective debate, for instance. This means that the same specific information can be seen for different persons and that a constant data interchange flow comes out. Also, it means that each member of the group can use it matching their personal goals inside this group or inside the course.

A second option allows a teacher to monitor the progress of his students, analysing dynamically all the contributions coming along the course and providing a proper feedback to them in both ways, one by one and collectively. This way, a bilateral information flow between learner and tutor is established, aimed to an academic and personalized good use. Level B provides the component 'monitor' and allows to view self properties and properties from the others in a structured way. These properties must be defined previous its use, and initialised if numeric data types are managed, and can be operated into formulas, as further is shown. Following, two examples about definition and initialisation:

```
<loc-property identifier="LP-LA-1-completed">
  <title>Response to the initial quiz</title>
  <datatype="boolean"/>
  <initial-value>true</initial-value>
</loc-property>
```

The property 'LP-LA-1-completed', with the sentence 'Response to the initial quiz' as a title or label, is a Boolean type and its initial value is set to TRUE. This variable is useful to store the current state of a learning activity to know whether it is finished.

```
<locpers-property identifier="LP-personalgoals">
```

```
<title>Which are your goals for this course?</title>
<datatype="text"/>
</locpers-property>
```

The property ‘LP-personalgoals’, with the title ‘Which are your personal goals for this course?’, is a Text type and it is not initialised so far. This variable is used to store personal information from the user and, therefore, there is no previous content. It is a local property and depends on every user.

If we want to use this last property (‘LP-personalgoals’) it can be read and saved along the normal course flow:

```
<Id:set-property ref="LP-personalgoals"/>
```

This code line allows to write (‘set’) some content in the property.

```
<Id:view-property ref="LP-personalgoals" property-of="self"/>
```

This code line allows to read (‘view’) the internal value of our own property (‘self’).

Moreover, this property can also be traced with the ‘monitor’ component. For instance, the following code line allows to read (‘view’) the property of a different student (‘supported-person’)

```
<Id:view-property property-of="supported-person" ref="LP-personalgoals"/>
```

To start this monitoring action, firstly the component ‘monitor’ must be set-up inside an environment (in this specific case):

```
<environment identifier="E-personalgoals">
  <title>Which are the goals of the others?</title>
  <service identifier="S-personalgoals">
    <monitor>
      <role-ref ref="Student"/>
      <title>Goals of the other students</title>
      <item identifierref="R-personalgoals"/>
    </monitor>
  </service>
</environment>
```

Above, the ‘monitor’ service is defined for a learner (‘Student’). This means that every student can view the content of his other classroom partners’ properties. In case a tutor needs to view students’ properties a similar structure can be written, providing a proper tracking of each participant in a course.

3.2 Adaptive learning and personalization

In order to illustrate this section we take the example ‘Learning to listen to Jazz’ (Tattersall and Burgos, 2005) initially developed in EML and later adapted to IMS LD. In this Unit of Learning a student can follow a course about Jazz and can choose two different itineraries, thematic and historic, based of his preferences. Also, some actions of monitoring can be tackled in the way described previously.

Adaptive learning (Shuell, 1988) pretends to choose, collect and show some contents to a student, coming from a common data base, and depending on the student’s initial profile and on the progressive results got during the running of the course. Personalization complements adaptive learning providing the capacity to choose specific features to each student, like content, look and feel, assessment and itinerary, all together inside a pre-made collective learning framework.

Regarding the concept of adaptive learning, and taking the referenced example, Jazz, a property called ‘LP-choose-itinerary’ is set-up to know whether the user has chosen one of the two itineraries. Each of them is described

inside their Activity Structure, 'AS-historic' and 'AS-thematic', previously defined in the manifest and out of the scope of this paper. All the process to choose an itinerary is programmed as a flow of conditions, taking one option or the another depending on the value of this property. Both Activity Structures are hidden in the beginning, when any value is inside the property yet:

```
<conditions>
<if>
  <no-value>
    <property-ref ref="LP-choose-itinerary"/>
  </no-value>
</if>
<then>
  <hide>
    <activity-structure-ref ref="AS-thematic"/>
    <activity-structure-ref ref="AS-historic"/>
  </hide>
</then>
```

The student can take one of the two options, 'historic' and 'thematic', available in a combo box. If the user takes the option 'thematic', his related structure 'AS-thematic' is shown and the non-related structure 'AS-historic' is hidden. The same things happens the other way around:

```
<if>
  <is>
    <property-ref ref="LP-choose-itinerary"/>
    <property-value>thematic</property-value>
  </is>
</if>
<then>
  <show>
    <activity-structure-ref ref="AS-thematic"/>
  </show>
  <hide>
    <activity-structure-ref ref="AS-historic"/>
  </hide>
</then>
```

These two different structures are able to have also non-identical contents or the same ones re-organized in several ways dealing with two complementary or opposite approaches and all is managed inside the same manifest coming with the Unit of Learning.

On the complementary concept, personalization, a simple case is to get the personal details of a student in the course. In the previous section we showed how to define and fill a property. Another possibility is to group several properties under a single name to operate easier. For instance

- a) defining a property, String type, and a second one, Integer type, and initialising this last one to zero

```
<locpers-property identifier="LP-name">
  <title>your name</title>
  <datatype datatype="string"/>
</locpers-property>
<locpers-property identifier="LP-age">
  <title>age</title>
  <datatype datatype="integer"/>
  <initial-value>0</initial-value>
</locpers-property>
```

- b) grouping both properties

```
<property-group identifier="LP-group-personalinfo">
  <title>personal details</title>
  <property-ref ref="LP-name"/>
  <property-ref ref="LP-age"/>
</property-group>
```

- c) requesting the related information to the user ('set') and showing the results ('view'), on demand

```
<ld:set-property-group ref="LP-group-personalinfo" property-of="self"/>
<ld:view-property-group ref=" LP-group-personalinfo" property-of="self"/>
```

3.3 Dynamic feedback and runtime tracking

In order to illustrate this section we take the example 'GeoQuiz' (Burgos, 2005). In this Unit of Learning a student answers a questionnaire consisting of five questions and a related additional feedback is given depending on the chosen responses. Later, a numeric final valuation is provided, meaning an average and a final remark based on it.

As a specific application on adaptive learning, IMS LD also allows to work with a) dynamic feedback, providing contents adapted to the students progress in a certain Learning Activity, and b) reading the results depending of his activity.

Regarding a) the provision of contents depending on a certain student evolution we have already seen a possibility in the previous example, making a call inside the manifest file itself to different structures with different, similar or equal contents and uneven orders. All in all, we are talking of showing and hiding an activity or a structure, while being related to resources, pointing themselves to external files.

Another possibility, though, deals with using classes to modify the visibility of specific content inside a external file pointed by a resource, this is, outside the manifest file. These classes are XHTML layers defined inside XML files besides the imsmanifest.xml file.

Although the mechanism that allows to hide and show these layers/classes looks like the already explained for Activity Structures, this action is now carried out in a file outside the manifest and, therefore, it runs rather different:

```
<if>
  <is>
    <property-ref ref="Answer1"/>
    <property-value>C</property-value>
  </is>
</if>
<then>
  <hide>
    < class="Feedback_Wrong"/>
  </hide>
  <show>
    <class="Feedback _Right"/>
  </show>
</then>
```

In this previous example we show the class 'Feedback_Right' only if the content of the answer 'Answer1' is equal to 'C' (this response is picking it up from an enumerated list), and hide the class 'Feedback_Wrong', with a very different content. In the external file, the value of the property 'Answer1' is set and both are defined, classes and their content:

- a) an answer is chosen ('set')

```
<p>Your answer is:
<set-property ref="Answer1" of="self"/></p>
```

- b) the classes are defined

```

<div class="Answer1_Wrong">
  <p><view-property ref="Answer1"/> is not right</p>
</div>
<div class="Answer1_Right">
  <p>Congratulations!</p>
</div>

```

Regarding b) the reading and interpretation of results depending on the user activity, we can make arithmetic calculations with the stored values along the course and provide a contextual feedback based on them. If we fancy two questions with two answers and two feasible values for each answer, 0 and 100, we could:

a) define every property

```

<locpers-property identifier="QuestionTrue1">
  <datatype datatype="integer"/>
  <initial-value>0</initial-value>
</locpers-property>

```

b) assign a value based on a specific answer

```

<if>
  <is>
    <property-ref ref="Answer1"/>
    <property-value>C</property-value>
  </is>
</if>
<then>
  <change-property-value>
    <property-ref ref="QuestionTrue1"/>
    <property-value>100</property-value>
  </change-property-value>
</then>
<else>
  <change-property-value>
    <property-ref ref="QuestionTrue1"/>
    <property-value>0</property-value>
  </change-property-value>
</else>

```

c) and, lastly, calculate a simple average with the two answers as arguments

```

<change-property-value>
  <property-ref ref="sum"/>
  <property-value>
    <calculate>
      <divide>
        <sum>
          <property-ref ref="QuestionTrue1"/>
          <property-ref ref="QuestionTrue2"/>
        </sum>
        <property-value>2</property-value>
      </divide>
    </calculate>
  </property-value>
</change-property-value>

```

3.4 ePortfolios and new forms of assessment

Traditional assessment grounds in the confrontation between the user knowledge against a machine or against the teacher knowledge. But it is not the only one way. New approaches of learning and teaching must come with new ways of assessment. Evaluation should be given along the learning flow and not just as an isolated resource a) checking the level of knowledge of a student before to decide his best itinerary, b) checking whether a concept is properly understood before stepping further with the next one and c) providing a high quality feedback to keep also a high motivation. Also d) scenarios consisting of several students must be taking into account sharing individual responses with the group for a common valuation and a collective debate, for instance (Koper and Tattersall, 2005). To formalize any of these proposals in LD is not difficult but hardworking, using the notes we have drawn through these pages. These four possibilities can be implemented like a combination of conditions and properties as long as formalizing evaluation questionnaires is feasible with IMS LD. Another challenge is too combine IMS LD and IMS Question and Test Interoperability Specification (IMS, 2003a). Although the implementation of QTI is outside the scope of this article, we can suggest how to integrate both specifications, issue that also concerns to LD. Basically, the sequence is the one already commented and it describes the relationship between an environment and a linked resource. The only difference is the type of the reference (will be a QTI type, like 'imsqti') that will link to a file written on this specification ('question_1.xml'):

```
<environment identifier="Env-1">
  <title>A test linking QTI and IMS LD</title>
  <learning-object identifier="LO-QTI-question-1">
    <title>First question</title>
    <item identifier="I-1" identifierref="R-que-1"/>
  </learning-object>
</environment>
```

```
<resource identifier="R-que-1" type="imsqti">
  <file href="question_1.xml"/>
</resource>
```

Lastly, the use of ePortfolios is feasible if the original drive of the specification IMS ePortfolio is held (IMS, 2002). It means to save and share data externally to any application and keep it standardized inside the information package, making possible that the information is consumed and managed in different places and systems. Global properties in IMS LD allow to define private information of an user (globpers-property), of a group (property-group) or, working as a constant, for everyone (glob-property). The first of these properties is also called 'portfolio-property' because allows to incorporate files to the Unit of Learning in runtime. As usual, this use needs two different moments:

- a) to create and set-up the global property

```
<globpers-property identifier="GP-suggestions">
  <global-definition uri="GP-suggestions ">
    <title>Suggestions about the course</title>
    <datatype datatype="file"/>
  </global-definition>
</globpers-property>
```

- b) to use the property in a XML file

```
<div class="upload-file">
  <p>Choose the file with your suggestions: </p>
  <ld:set-property ref="GP-suggestions"/>
</div>
```

4. Conclusion

Although Level A in IMS LD provides the base of any Unit of Learning and draws the skeleton of any learning flow, Level B provides several mechanisms to improve and make this moved lesson plans stronger. Properties,

conditions, global elements and monitoring services are four main features of the specification that make Level B the most powerful one. This means that the learning scenarios could include important practical pedagogical uses and aim to be modelled in IMS LD, mainly using the Level B layer. Active learning, collaborative learning, adaptive learning, personalization, dynamic feedback, runtime tracking, ePortfolios and new ways of assessment are all of them key teaching and learning resources that we point out how to model in IMS LD. Bearing this in mind it is fair to say that, in the current state in the development of tools, it is not a trivial issue to move the pedagogical approach to a running Unit of Learning. Many of the current tools are technically built but not pedagogically gifted. Users still need to know some of the specification to build Level A units and quite a lot of XML coding to create Levels B and C units. In practice, it means that nowadays is not so easy for a non-technical end-user to edit and run Units of Learning in actual lesson plans.

The most recent discussions on IMS LD tooling point to design an on-top layer more focused on usability, pedagogical and navigational issues more than on the technical ones. Improving the message and the metaphor in the editor and in the player to work with will certainly enrich the view of the teachers and lesson plans builders and make their work easier and faster. This way, teachers and learning designers could build Units of Learning bottom-up (focused on their experience and needs) instead of up-bottom (focused on understanding the specification itself), using an inductive approach.

5. References

1. Bolton (2004) *Reload Project*. United Kingdom: The University of Bolton, The University of Strathclyde and JISC. Available at <http://www.reload.ac.uk/editor.html>. Retrieved August 31st, 2005
2. Burgos, Daniel (2005) Geo Quiz Example Unit of Learning. Available at <http://moodle.learningnetworks.org>. Retrieved on September 29th, 2005
3. Cole, M., Engestrom, Y. (1993) *A cultural-historical approach to distributed cognition*, In G. Salomon (Ed.), *Distributed cognitions*. New York: Cambridge University Press
4. Dougiamas, M. (2004) *Moodle*. Available at <http://moodle.org/>. Retrieved on August 31st, 2005
5. IMS (2002) *ePortfolio*. Boston: USA. Available at www.imsglobal.org. Retrieved on July 29th, 2005
6. IMS (2003) *Learning Design*. Boston: USA. Available at www.imsglobal.org. Retrieved on July 30th, 2005
7. IMS (2003) *Question and Test Interoperability*. Boston: USA. Available at www.imsglobal.org. Retrieved on July 29th, 2005
8. Koper, R., Tattersall, C. (2005) *Learning Design: A Handbook on Modelling and Delivering Networked Education and Training*. Germany: Springer Verlag
9. OUNL (2004). *Learning Network for Learning Design*. Heerlen: Open University of The Netherlands, OTEC. Available at <http://moodle.learningnetworks.org>. Retrieved on August 14th, 2005
10. OUUK (2005) Sled player. Open University of United Kingdom. Available at <http://sled.open.ac.uk>. Retrieved on September 29th, 2005
11. Shuell, Th.J. (1988) The role of the student in learning from instruction, *Contemporary Educational Psychology*, 13, 276-295
12. Tattersall, Colin and Burgos, Daniel (2005) Learning to Listen to Jazz Example Unit of Learning. Available at <http://moodle.learningnetworks.org>. Retrieved on September 29th, 2005
13. UNFOLD (2004) *UNFOLD Project*. Available at www.unfold-project.net. Retrieved August 30th, 2005
14. Van der Vegt, Wim (2005) *CopperAuthor*. Heerlen: Open University of The Netherlands. Available at www.coppercore.org. Retrieved on July 29th, 2005
15. Vogten, H., Martens, H. (2005) *CopperCore 2.2.4*. Heerlen: Open University of The Netherlands. Retrieved at www.coppercore.org. Retrieved on October 9th, 2005