Developing and using an IMS LD ontology to create and execute learning designs

EDUARDO SÁNCHEZ, MANUEL LAMA and RICARDO R. AMORIM
Department Electronics and Computer Science
University of Santiago de Compostela
School of Physics, 15782
SPAIN

Abstract: - This paper describes how ontologies can be used to model, validate and execute IMS Learning Design. Concretely, we propose an IMS LD ontology, which covers both IMS LD level A and level B, and explain how it was developed and then used during the IMS LD document life cycle. The main contribution can be summarized as follows: (1) the formalization of the implicit knowledge found on the IMS LD specification in terms of formal axioms, (2) their use to validate IMS LD documents, and (3) a learning design translator to translate representations between XML Schema and F-Logic.

Key-Words: - IMS LD Specification, modeling languages, ontologies

1 Introduction

The specifications for the learning design, known as Educational Modelling Languages (EML), are models of semantic information and aggregation that describe, from a pedagogic point of view, the content as well as the educational activities. These elements are organized into units of study with the aim of allowing their reuse and interoperability [1]. Moreover, EMLs facilitate the description of pedagogic aspects that are related with LOs in educational processes [2].

The IMS Learning Design specification (IMS-LD) [3] is an Educational Modelling Language (EML) that provides a model of semantic notation to describe both the content and processes of units of study. This specification, drawn up by the IMS/LDWG work group, is an integration of the EML developed by the OUNL (Open University of Netherlands), with other existing IMS specifications for the exchange and interoperability of e-learning material. The OUNL EML is a meta-vocabulary that is defined based on the diversity of concepts existing in a wide range of pedagogic techniques. The IMS EML incorporates the OUNL EML, and describes the structure and educational processes based on a pedagogic metamodel, using units of learning called Learning Design. IMS LD describes a method that is made up of a number of activities carried out by both learner and staff in order to achieve some learning objectives. It allows the combination of various techniques (traditional, collaborative, etc.), and facilitates the description of new ones. From the proposed specifications, the IMS LD has emerged as the de facto standard for the representation of any learning design that can be based on a wide range of pedagogical techniques.

The IMS LD information model specifies three levels of modelling: (1) level A, containing the core concepts needed to model any pedagogical situation; (2) level B, extending the level A with Properties and Conditions, aimed at supporting more sophisticated behaviours enabling personalization and learner interactions, and (3) level C, adding Notifications to the previous levels. With these elements, the IMS LD offers flexible expressiveness possibilities to represent the design as well as an XML Schema representation to execute IMS LD documents during runtime.

In this paper, we address the issues regarding with the modeling and execution of Learning Designs. Section 2 introduces some of the problems related with the modeling and execution of learning designs. Section 3 proposes the use of ontologies to solve these problems. Section 4 introduces the IMS LD ontology. Section 5 deals with the development of the ontology. Section 6 explains how it can be used during the IMS LD life cycle. Section 7, finally, draws the conclusions of this work.

2 Issues on modeling and executing Learning Designs

To facilitate the interoperability between software systems, the IMS LD specification has been formally
modelled through the XML-Schema language [4,5]. However, the knowledge model of this language is not expressive enough to describe the semantics (or meaning) associated to the elements of the IMS LD. Thus, the main limitations of the XML-Schema language are [6]:

- Hierarchical (is-a) relations between two or more concepts cannot be explicitly defined.
- Properties of relations cannot be defined.
- General and formal constraints (axioms) between concepts, attributes, and relations cannot be specified.

IMS LD is also intended to execute instances of IMS LD documents during runtime. To accomplish this, the logical architecture of an IMS LD runtime system should consist of three main modules (Authoring, Production and Delivery) that have to solve the following issues [7]:

1. Authoring. It deals with modelling and creation of IMS LD documents
2. Production. It deals with (a) validation, (b) publication, and (c) population of IMS LD documents.
3. Delivery. It deals mainly with personalization and role population issues.

Most of these tasks could be automated by using software agents working on behalf of IMS LD authors, publishers, and learners. But in order to fully use the capabilities of those agents, the knowledge regarding with the learning design process has to be described in a formal and explicit way.

3. The use of Ontologies to model the IMS LD Specification

Ontologies [8] come handy to describe formally and explicitly the structure and meaning of the metadata elements; that is, an ontology would semantically describe the metadata concepts. In the educational domain, several ontologies have been proposed: (1) to describe the learning contents of technical documents [9]; (2) to model the elements required for the design, analysis, and evaluation of the interaction between learners in computer supported cooperative learning [10]; (3) to specify the knowledge needed to define new collaborative learning scenarios [11]; or (4) to formalize the semantics of learning objects that are based on metadata standards (like LOM) [12]. The focus of that research is either on the development of a taxonomy of concepts on the basis of a established theory or specification (1 to 3), or on the formal definition of the metadata using an ontology language (4). However, none of them deal with the formal description of the meaning of the concepts, and they do not address the ontological modelling of any specification for learning design.

Ontologies are then required (1) to formally describe the semantics implicit in the information model, and (2) to enable software agents to perform complex tasks. Some examples could be:

1. Authoring Stage. A set of axioms could be used by authoring agents to guide the creation of IMS LD documents, as well as to validate their logical consistency.
2. Production Stage. The ontology could be used by production agents to automate different tasks such as: resource allocation based on availability, creation of new IMS LD runs based on the number of registered students, role assignments based on student profiles, and so on.
3. Delivery Stage. The ontology could also be used to solve the problem of personalization or adaptation of the learning design to the individual users. It involves different tasks such as the selection of suitable learning activities, contents, etc.

4. The IMS LD Ontology

To develop the Learning Design ontology we have created a concept taxonomy (Figure 1), which describes the elements of the IMS LD conceptual model and the IMS LD information model, and a set of axioms (Table 1), which formally constraint the semantics of the concept taxonomy on the basis of the explanations formulated in natural language in both information and behavioural models [13].

The concepts were obtained from elements and relations explicitly represented in the IMS-LD information model document. A set of axioms was additionally included to guarantee the consistency of the definitions and to represent implicit knowledge. The axioms were obtained from restrictions identified after an extensive analysis of the explanations, written in natural language, provided by the IMS-LD Information Model and the Best Practice and Implementation Guide documents.
In figure 1, the Unit of Learning concept (UoL) integrates the description of the Resources and the Learning Design (LD). The Resource concept allows representing various entities, like physical resources, and concepts whose attribute description is domain-dependent.

The Learning Design concept is related to the Learning Objective (define the intended outcomes) and the Prerequisite (the previous knowledge needed) concepts. These concepts are subclasses of Item and they can be mapped onto the equivalent concepts in the Resource hierarchy. The Components (Role, Execution Entity and Environment) of the Learning Design represent pieces of the educational process. Role instances participate in an Execution Entity in the context of an Environment.

The IMS-LD level B adds the Property concept to the level A to extend the description of the elements of level A. Properties represent data that can be stored enabling to maintain information about Role, Learner and the execution state of some entities such as Play, Act, Activities. The properties constitute an essential part of monitoring, personalization, assessment, and user-interaction processes.

Level B also adds the Conditions concept (not shown), which is aggregated into the Method concept and it is used along with properties to facilitate the learning design refinement. For example, in a Method, Conditions can be introduced either to show or hide entities such as Play, Activity and Environment in a personalized way. The Conditions concept has a typical programming language structure: IF [expression] THEN/ELSE [show/hide something or change property-value].

### Table 1. Example of axioms: Level A (1 and 2), and Level B axioms (3)

<table>
<thead>
<tr>
<th></th>
<th>IMS LD Specification</th>
<th>Explanation</th>
<th>Formal Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Inf-Mod, Page 83 (item 17): “Environments are connected to activities, activity-structures or roles (in a role-part). When an activity-description is visible, always the connected environment (including the content structure of the environment) must be made visible. It must be possible to access and see the activity-description and the content of one of the objects or services within the environment at the same time.”</td>
<td>When the value of the invisible attribute of an Activity Description is “true”, the value of that attribute for the Environments connected to the Activity associated to the Activity Description must be also true.</td>
<td>∀ a, ad, e, lo, s</td>
</tr>
<tr>
<td>2</td>
<td>Inf-Mod, Page 40 (item 0.5.1): “This element states that a play is completed when the last act is completed.”</td>
<td>The Act referred as the value of the attribute when-last-act-completed of a Play must be one of the Acts associated to the Play.</td>
<td>∀ p, cp, a</td>
</tr>
<tr>
<td>3</td>
<td>Inf-Mod, Page 51: “Property values may be calculated from the values of other properties. It is also possible to take over the property value of another property (with property-ref).”</td>
<td>In a Change-Property-Value, if the related property value was obtained from the value of another property, both, the datatype of the property to be changed and the datatype of the other property must be the same.</td>
<td>∀ cpv, p, pl, pv</td>
</tr>
</tbody>
</table>

![Figure 1. Main elements of the concept taxonomy: Level A (white) and Level B concept (grey)](image-url)
5 Developing the ontology

For the construction of the ontology we have used Protégé, an extensible, platform-independent environment for creating and editing ontologies and knowledge bases [14]. As it is shown in Figure 2, after describing the ontology at the knowledge level, the Protégé-OWL plug-in was used to export it in the OWL language [15], which is the W3C recommendation for the Semantic Web.

![Fig 2. Building the Ontology with Protégé](image)

6 Developing and executing learning designs with the IMS LD Ontology

The steps to create and execute the Learning Design (LD) document are illustrated in Figure 3. The LD document is initially created with Reload and represented in XML Schema. After that, the XML document is translated to F-Logic, the language used to represent the ontology concepts and their instances, by means of a Learning Design Translator. For the translation, FLORA-2, an object-oriented language for building knowledge-intensive applications, was used. As the LD document is represented in F-logic, its consistency is checked by taking advantage of the axioms of the ontology. Thereafter, a corrected LD document in XML Schema is generated and finally executed in CopperCore.

The IMS LD document could also be initially created with the Protégé tool. After that, a translation from this code into the IMS LD XML-Schema representation would be carried out. Finally, this representation could be imported in the Reload tool as a learning design. Once the IMS LD XML-Schema representation is obtained, the Reload tool can be used to test whether the learning design is IMS LD compliant or not. As Figure 4 shows, the result of this test is that the learning design entities specified in Protégé (and, equivalently, in OWL) are correctly identified and represented in the Reload tool. Therefore, the LD ontology is compliant with the IMS LD XML-Schema specification.

In this context, the LD ontology could be also used in a learning management system that implements the IMS LD specification following the XML-Schema language. In such case, specific middleware to translate from the ontology (expressed in F-LOGIC or OWL) into the XML-Schema representation, and vice versa, would be required. For instance, in a service oriented architecture, the translation operation could be offered by a Web service that would receive SOAP messages whose content is (part of) the ontology to be translated, and would send the result of the translation using the same protocol.

7 Discussion

IMS LD Specification is an excellent and mature starting point to create LD documents. However, complementary approaches, such as ontologies, could be useful to solve the semantic issues derived from the XML representation of the IMS LD Specification. The most notably contribution is the formalization of a set of axioms that represent relevant knowledge expressed informally in natural language. Moreover, the development of learning design translators facilitates the validation of IMS LD documents while maintaining the Reload tool to create the documents.
Figure 3. LD document life cycle with the IMS LD ontology

Figure 4. Translation of a learning design document from OWL to XML-Schema
References:


