Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005

Ontology of Learning Object Content Structure

Jelena JOVANOVIĆ, Dragan GAŠEVIĆ
FON – School of Business Administration, University of Belgrade
Jove Ilica 154, 11000 Belgrade, Serbia & Montenegro

Katrien VERBERT, Erik DUVAL
Dept. Computerwetenschappen, Katholieke Universiteit Leuven
Celestijnenlaan 200A, B-3001 Leuven, Belgium

Abstract. This paper proposes an ontology that enables a formal definition of Learning Object (LO) content structure. The ontology extends the Abstract Learning Object Content Model (ALOCoM) with concepts from information architectures. It defines a number of concepts that represent different types of content units and it specifies their structure. Formalising structural aspects of LOs, the ontology facilitates re-purposing of LOs at different levels of content granularity, i.e. LOs in their entirety and their components. Furthermore, being a generic LO content model, the ontology serves as an integration point of heterogeneous LO content models.

Introduction

There is an increasing interest in the learning technology community for repurposing learning objects (LOs) [1]. Presently, authors of learning materials employ a cut & paste approach when composing new LOs out of components of existing ones. Nonetheless, such an approach is non-scalable in terms of maintenance, since each time you copy a content unit, you create a new place that needs to be maintained [2]. Additionally, the process tends to be error-prone, and due to its inherent monotony, easily becomes both bothering and time consuming. The authors are in a much better position if access to the components of LOs and their composition into meaningful units is made, at least partially, automatic. A possible solution employs a more reusability prone format of LOs that makes their structure explicit and thus enables reusability of LO components as well. This can be accomplished through provision of a flexible model of LO content structure. An explicit content structure allows the disaggregation of a LO into its constituent components. Those components, enriched with fine-grained descriptions (metadata), increase the findability of relevant content units.

Ontologies and Semantic Web technologies can be a solid basis for solving the aforementioned problem, as an ontology gives a formal specification of the shared conceptualization of a certain domain. For the domain of e-learning, we found a classification of ontologies suggested in [3] relevant. The classification differentiates between: a) *content* (*domain*) ontologies describing the subject domain of a content unit, b) *context* (*didactic*) ontologies formally specifying the educational/pedagogical role of a content unit, c) *structure* ontologies providing a shared conceptualization of how content units can be assembled together to form a coherent learning whole.

High level of LO re-purposing can be achieved if learning materials are broken down into small content units that can be easily handled. Accordingly, concepts from the structure ontology are especially useful. If we have LO repositories with learning content disaggregated to content units of the lowest level of granularity (e.g. a single image, text fragment or audio/video clip) and presented in a structure ontology-aware format, we will be able to make the process of composing new learning materials out of components of existing LOs (partially) automatic. Furthermore, this structure related information would also be of great importance to a dynamic assembly engine of an Adaptive Learning System when combining content units into a meaningful and well structured learner tailored presentation.

In this paper, we present an ontology that we propose for the formal specification of LO content structure. The ontology extends the Abstract Learning Object Content Model (ALOCoM) that defines a framework for LOs and their components [4], with concepts from the Darwin Information Typing Architecture (DITA) – an XML-based architecture for authoring, producing, and delivering technical information that is easy to reuse [2].

The paper is organized as follows: in the next section we give a concise overview of the conceptual origins of the ALOCoM ontology and we briefly describe the ontology architecture. In the second section we explain the ontology implementation in detail. Section 3 explains the enabling role that the ontology has in achieving interoperability among different content models and Section 4 concludes the paper.

1. Conceptual Solution

This section explains the conceptual origins of the ontology, thus enabling easier comprehension of the ontology architecture and design.

The Ontology Origins

As we stated in the introduction, the proposed ontology is a generic content model that defines a framework for LOs and their components [4]. As Figure 1 suggests, the model differentiates between Content Fragments (CF), Content Objects (CO), and Learning Objects (LO).

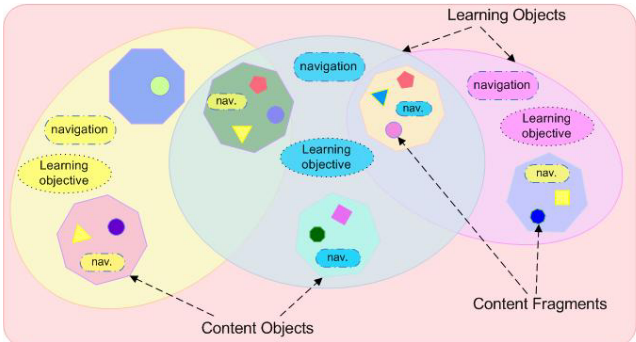

Figure 1. A sketch of Abstract Learning Object Content Model

CFs are content units in their most basic form, like text, audio and video. Basically, CFs are raw digital resources. They can be further specialized into discrete (graphic, text, image) and continuous (audio, video, simulation and animation) elements. COs aggregate CFs and add navigation. Navigation elements enable proper structuring of CFs within a CO. Besides CFs, a CO can include other COs as well. At the next aggregation level, a LO is defined as a collection of COs with an associated learning objective.

Further, we defined content types for each of these components. We introduced CF types such as image, text, audio and video. For defining CO types, we investigated existing Information Architectures, like the Information Block Architecture [5] developed by Dr. Horn and the IBM Darwin Information Typing Architecture [6]. These architectures define information types (e.g. concept, principle, task) and their building blocks (e.g. example, definition, analogy). As a starting point, we defined the CO types and their structure using DITA concepts, since DITA is a recent architecture with rich documentation and online support [6]. Besides CF and CO types, the ontology identifies LO types such as a Lesson, a Report, a Course and a Test. Finally, the ontology defines the relationships between the LO components. For now, aggregational and navigational relations are specified.

1.2 The Ontology Organization

An important feature of the DITA architecture is the extensibility of the core information types aimed to meet specific needs of an author/community. Since our objective is to have a content structure ontology that supports different kinds of LOs, and that is easily extensible to include new LO types, we decided to make use of DITA's inherent extensibility in the ontology we were developing. Therefore, we organized the ALOCoM ontology as an extensible infrastructure consisting of: the core part (ALOCoMCore) with concepts common for all LO types and an unlimited number of extensions, each extension supporting one specific LO type. Figure 2 illustrates this hierarchical architecture. The main benefits of the proposed, extensible, ontology architecture is to avoid large and clumsy vocabularies: ontology extensions can meet specific requirements of each application domain. In other words, exclusively the ontology extension defined for a specific LO type that the application works with, should be included to avoid unnecessary information burden.

Additionally, the core part of the ALOCoM ontology is an integration point of different LO content models (SCORM, CISCO, Learnativity, etc.). Therefore, we defined extensions of the core ontology that serve as mappings between ALOCoM and other LO content models. This topic is further extended in the section 3.

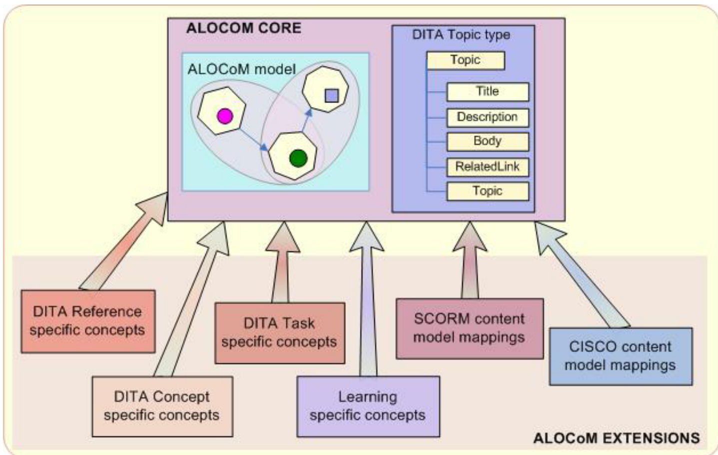

Figure 2. A vision of hierarchical structure of the ALOCoM ontology

2. The Ontology Implementation

We used the Web Ontology Language (OWL) – the W3C recommendation [7] – to develop the ALOCoM ontology and exploited advantages of OWL specific features for ontology development. These features can be summarized as follows:

- Solid modularization mechanism that enables the definition of easily extensible ontologies.
- Support for definitions of concept hierarchies, so that reasoners can recognize the presence of the inheritance (is-a) relationship between two concepts.
- Advanced ways for describing properties like: the range of a property defined as a union of two or more other classes, definition of cardinality restriction, etc.
- Ability to define synonyms, so we can make equivalences (or mappings) between the concepts of two (or more) vocabularies covering the same domain. For example, we can define mappings between ALOCoM and SCORM terminology – e.g. an ALOCoM CF is equivalent to a SCORM Asset.

To implement the ontology, we used the Protégé ontology development tool (http://protege.stanford.edu), since it has support for development, storage and editing of ontologies in OWL format.

In the following subsections we present the ontology in detail. First, we explain the design of the core part of the ontology and then focus on the ontology extensions.

2.1 The Core Ontology

The first step in building the core part of the ontology was to define classes for representing CFs, COs, and LOs in general. Subsequently, we added a number of classes corresponding to the specific types of a LO components (i.e. COs and CFs).

As we stated in section 1.1, the ALOCoM ontology defines a number of CF types divided into two main categories of continuous and discrete CFs. Accordingly, we extended the *ContentFragment* class of the ontology with *ContinuousCF* and *DiscreteCF* classes, respectively representing these two main CF types. The *DiscreteCF* is further specialized into *Text*, *Image* and *Graphic* classes, while the *ContinuousCF* is further extended with *Audio*, *Video*, *Animation* and *Simulation* classes.

Further, we extended the *ContentObject* class of the core ontology with a number of classes representing different kinds of COs that can be part of almost any type of LO. We based those classes on elements of the DITA information architecture. One ontology class is introduced for each DITA element that we found appropriate for describing content units typical for the learning domain. Accordingly, many of the DITA building blocks, such as *section*, *paragraph*, *list* etc., are included in the core ontology as either direct or indirect subclasses of the *ContentObject* class. We did not include those DITA elements that are presentation-related, such as the *searchtitle* element that is used when transforming a DITA element to XHTML to create a title element at the top of the resulting HTML file [6].

One should note that, even though the ALOCoM ontology is based on the DITA model, some of the ontology concepts are not identical in meaning to the corresponding DITA elements. The primary reason for this lies in the obvious discrepancy of the intended application domains of DITA and ALOCoM: while DITA is devised exclusively for the technical domain, the ALOCoM ontology is intended to be used in a variety of learning domains. Therefore, we need to make the structure of certain DITA elements more general, so that they can be applicable not just for structuring of technical information, but also for structuring of content in any other learning domain (e.g. mathematics, arts, etc). Additionally, the structure of certain DITA elements is overwhelmed with presentation-related components (e.g. *table*, *link*, *definitionlist*). Being interested in content structure released from presentation details, we created ontology classes corresponding to a simplified version of such DITA elements (e.g. Link, Definition), leaving out all of their presentation-oriented components. Generally speaking, DITA served us as a good starting and reference point to get an overview of the concepts potentially relevant for an explicit specification of LO structure.

The *LearningObject* class is introduced to represent the LO content type. Descendents of this class are defined in the ontology extensions. Each extension typically covers one specific LO type.

Finally, the core part of the ALOCoM ontology defines several types of properties. From the perspective of content structuring, the following four are the most important: *hasPart*, *isPartOf*, and *ordering*. The definition of these properties is graphically represented in Figure 3, using the Ontology UML Profile – OUP presented in [8].

The *hasPart* and its inverse *isPartOf* properties allow us to express aggregational relationships between content units. The domain of the *hasPart* property is defined as the union of COs and LOs, since CFs represent elementary content units that cannot be formed of smaller meaningful content units. The range of this property is defined as the union of CFs, COs and LOs. We exploited the mechanism of restrictions to constrain the range of this property for almost each type of both COs and LOs. For example, in the case of the *List* CO type, the range of this property is restricted to encompass only instances of the *ListItem* type, or in the case of the *Table* CO type, the range of the same property is restricted to the union of *TableRow*, *TableData* and *Title* classes. Similar restrictions are defined for the *isPartOf* property. In the left part of Figure 4, we used OUP to depict restrictions imposed on the range of the *isPartOf* property in the context of the *ListItem* concept. As the figure shows, the range of the property is limited solely to the instances of the *List* class. The right part of the same figure presents the diagram in the OWL XML binding.

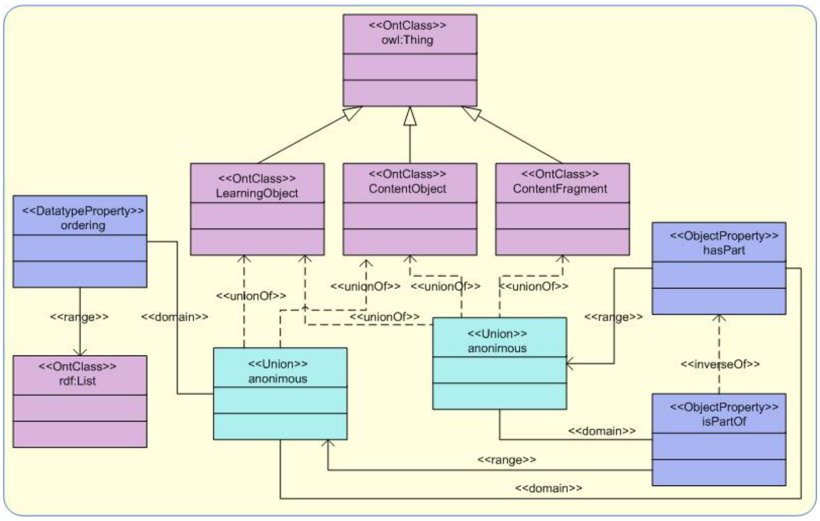

Figure 3. A scatch of major properties of the ALOCoM ontology in OUP

The *ordering* property allows us to express sequencing of components aggregated in a composite content unit (e.g. sequencing of CFs inside a CO). The domain of this property is a union of COs and LOs. CFs are not included in the domain, since CFs are elementary content units that cannot be further dissagregated. The range of this property is an rdf:list consisting of identifiers of components belonging to the composite content unit. The order of these identifiers in the rdf:List defines the order of components in a composite content unit (i.e. CO or LO). The elements of such an rdf:List must be identifiers of the resources that form the range of the *hasPart* property of the composite content unit. A composite content unit can have an arbitrary number of ordering properties, each one defining a specific learning path.

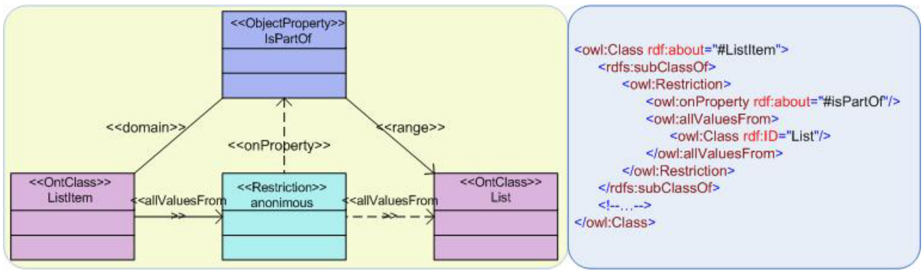

Figure 4. Restriction on the range of the *isPartOf* property of the *ListItem* class

2.2 The Ontology Extensions

As it was previously stated, the ALOCoM ontology is organized as an extensible architecture. Each extension of the core part of the ontology introduces a set of classes representing content units specific for a certain content type. Up till now, we defined three extensions, namely ALOCOMConcept, ALOCoMTask and ALOCoMReference, each one corresponding to a DITA core information type (*concept*, *task* and *reference* respectively). Due to the space limit, we shall briefly describe just one of those extensions, ALOCoMTask. Within this extension, we introduce classes corresponding to the content units specific for the DITA *task* information type. *Task* generally provides step-by-step instructions explaining how to perform certain task, i.e. what to do and in which order [6]. In Figure 5 the ontology classes introduced in this extension are presented in violet (*Task*, *TaskContext*, *TaskPrereq*, *TaskPostReq*, *TaskBody*, *Info*, *Command*, *Choice*, *Step*, *Result*), while concepts from the core ontology are in dark blue (*owl:Thing*, *LO*, *CO*, *Topic*, *Body*, *CF*).

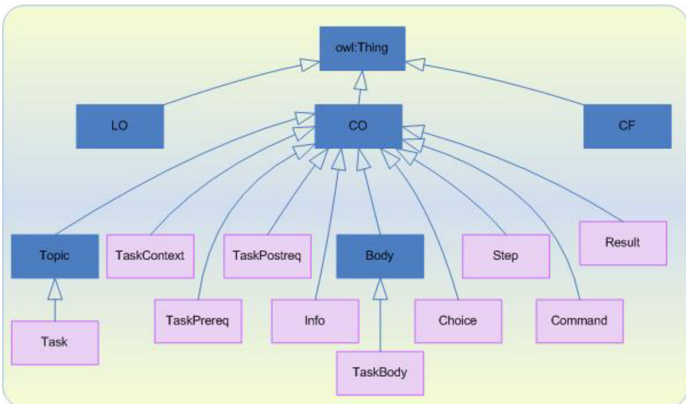

Figure 5. ALOCoMTask ontology extension

Since our intention is to enable content structuring in the learning domain, we are naturally interested in enriching the ontology with additional classes representing content units common to learning situations. Therefore, we are currently developing an extension, named ALOCoMLearning (Figure 6). We introduced, among others, a question, answer and exercise building block, since these content units are typical for learning. DITA does not provide these building blocks as the intent of DITA is primarily technical documentation. Furthermore, classes such as *Lesson*, *Test* and *Course* are defined as new types of LOs.

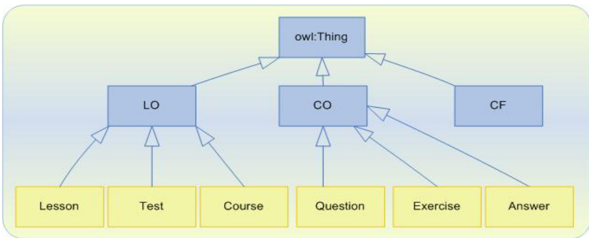

Figure 6. ALOCoM ontology extension with learning-specific classes

3. Ontology-based content model mappings

The semantic heterogeneity of LO content models (e.g. a SCORM Asset is equivalent to a CISCO Content Item) prevents us to automate the process of assembling a new LO out of content units defined in compliance with different content models. Accordingly, there is a need for a generic LO content model that would enable reuse and repurposing of content units developed according to one content model in the context of another one. The ALOCoM ontology, being built on such a generic model, has a potential to serve as a mediator, enabling communications between disparate LO content models.

We base our approach on a method proposed in [9] for integrating data using ontologies. The method has three main stages: building a shared vocabulary, building local ontologies and defining mappings. We have developed the ALOCoM ontology that has the role of a shared vocabulary, as well as one (local) ontology for each investigated LO content model (SCORM, CISCO, Learnativity, NCOM, NETg) and we defined mappings between the global and local ontologies. Table 1 gives a rough overview of those mappings. The next step is to implement those mappings so that resoners can use them to perform automatic translations between different content models. Since both global and local ontologies are written in OWL, we used the *owl:equivalentClass* property to express semantic equivalences between concepts from the global and local vocabularies. However, mappings implemented in such a way are sufficient for some simple reasonings, but in some situations we would need a more expressive mechanism [10]. Therefore, we are considering using RuleML (http://www.dfki.uni-kl.de/ruleml/) or the Semantic Web Rule Language – SWRL [11], as declarative languages for expressing rules, in this case transformation rules. An alternative would be to use a Java-based framework for the Semantic Web (e.g. Jena, http://jena.sourceforge.net/) that provides a Java API for working with ontologies.

Table 1. An overview of mappings between analyzed LO Content Models and ALOCoM

<table>
<tr><td>ALOCoM</td><td>Content Fragment</td><td>Content Object</td><td colspan="4">Learning Object</td></tr>
<tr><td rowspan="3">Learnativity</td><td rowspan="3">Raw Media Element</td><td rowspan="3">Information Object</td><td colspan="4">Application Specific Object</td></tr>
<tr><td colspan="4">Aggregate Assembly</td></tr>
<tr><td colspan="4">Collection</td></tr>
<tr><td>SCORM</td><td>Asset</td><td>Sharable Content Object</td><td colspan="4">Content Aggregation</td></tr>
<tr><td rowspan="3">CISCO</td><td rowspan="3">–</td><td>Content Item</td><td colspan="2" rowspan="3">Reusable Information Object</td><td colspan="2" rowspan="3">Reusable Learning Object</td></tr>
<tr><td>Practice Item</td></tr>
<tr><td>Assessment Item</td></tr>
<tr><td>NETg</td><td>–</td><td>–</td><td>Topic</td><td>Unit</td><td>Lesson</td><td>Course</td></tr>
</table>

4. Conclusions

In this paper, we presented the ALOCoM ontology that we developed to provide a more explicit specification of the structure of learning content units. With such an ontology we are able not only to reuse complete learning units, but also to reuse their components. To build the ontology we used some concepts form the DITA architecture, while we adapted some of them to better support the e-learning domain. The ALOCoM ontology is organized as an extensible architecture comprising one core part with the concepts common for all LO types and an unlimited number of extensions for each supported LO type. Apart from defining the common concepts in the ontology core, we defined semantic equivalencies between the ALOCoM ontology and several well-known content models (e.g. SCORM, CISCO, etc.).

We regard the ontology as a promising starting point for our further research towards achieving automated mappings between the most important content models as well as different LO types. We are currently setting up an ALOCoM ontology based LO repository and framework [12] that we are going to use for performing experiments on the ontology. Our goal is to evaluate to what extent the ontology can be used as a mediator for bridging different content models. We are also planning to extend the ontology by using some of Semantic Web rule languages (e.g. RuleML) in order to have more precise mappings between ALOCoM and other content models.

References

[1] Duval, E. and Hodgins, W., "A LOM research agenda", *In Proceedings of the 12th International World Wide Web Conference*, Budapest, Hungary, 2003, pp.1-9.

[2] Priestley, M., "DITA XML: A Reuse by Reference Architecture for Technical Documentation", *In Proceedings of the 19th Annual International Conference on Computer Documentation,* ACM SIGDOC 2001, Santa Fe, New Mexico, USA, October 21-24, 2001. pp. 152-156.

[3] Stojanovic, Lj., Stabb, S. and Studer, R., "eLearning based on the Semantic Web," *In Proc. of the WWWNet Conf.*, Orlando, USA, 2001.

[4] Verbert, K. and Duval, E., "Towards a global architecture for learning objects: a comparative analysis of learning object content models," *In Proc. of the 16th ED-MEDIA 2004 Conf.*, Lugano, Switzerland, 2004, pp. 202-209.

[5] R. E. Horn. Structured writing as a paradigm. In Instructional Development: the State of the Art. Englewood Cliffs, N.J., 1998.

[6] DITA Language Reference. Release 1.2. First Edition, May 2003. http://xml.coverpages.org/DITALangugeReference20030606.pdf.

[7] Bechhofer, S., et al (2004) "OWL Web Ontology Language Reference," W3C Recommendation, http://www.w3.org/TR/2004/REC-owl-ref-20040210.

[8] Djurić, D, Gašević, D., Devedžić, V., "Ontology Modeling and MDA," *Journal of Object Technology*, Vol. 4, No. 1, 2005, pp. 109-128.

[9] Buccella, A., Cechic, A. and Brisaboa, N.R. (2003). "An Ontology Approach to Data Integration," *Journal of Computer Science and Technology*, Vol.3 No.2, pp. 62-68.

[10] Hatala, M. and Richards, G., "Value-added Metatagging: Ontology and Rule-based Methods for Smarter Metadata," *In M. Schroeder and G. Wagner (Eds.) Rules and Rule Markup Languages for the Semantic Web (RuleML2003)*, LNCS 2876, Springer-Verlag, pp.65-80, 2003.

[11] I. Horrocks, P.F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean. (2004). SWRL: A Semantic Web Rule Language Combining OWL and RuleML, version 0.5 of 19 November 2003. [Online]. Available:http://www.daml.org/2003/11/swrl/rules-all.html

[12] Verbert, K., Jovanović, J., Gašević, D., Duval, E., and Meire, M., "Towards a Global Component Architecture for Learning Objects: a Slide Presentation Framework", In Proc. Of the *17th ED-MEDIA 2004 Conf.*, Montreal, Canada, 2005. (to appear).