INTEGRATING SEMANTIC WEB REASONING INTO LEARNING OBJECT METADATA

Shang-Juh Kao, I-Ching Hsu
Department of Computer Science, National Chung-Hsing University, Taiwan

Keywords: LOM, Semantic Web, SCORM, OWL.

Abstract: One of important functions of Learning Object Metadata (LOM) is to associate XML-based metadata with learning objects. The inherent problem of LOM is that it's XML specified, which emphasizes syntax and format rather than semantic and knowledge representation. Hence, it lacks the semantic metadata to provide reasoning and inference functions. These functions are necessary for the computer-interpretable descriptions that are critical in the reusability and interoperability of the distributed learning objects. This paper aims at addressing this shortage, and proposes a multi-layered semantic framework to allow the reasoning and inference capabilities to be added to the conventional LOM. To illustrate how this framework work, we developed a Semantic-based Learning Objects Annotations Repository (SLOAR) that offers three different approaches to locate relevant learning objects for an e-learning application - LOM-based metadata, ontology-based reasoning, and rule-based inference.

1 INTRODUCTION

The Sharable Content Object Reference Model (SCORM) (SCORM, 2004) is developed and extended based on IEEE Learning Object Metadata (LOM) (IEEE LOM, 2002). The LOM in SCORM is used to describe SCORM-compliant learning objects in a consistent fashion such that they can be identified, categorized, searched for and discovered within and across systems to further facilitate sharing and reuse.

The inherent problem of LOM is that it is based on XML, which lays stress on syntax and format rather than semantic and knowledge representation. Hence, LOM exhibits the advantage of data transformations and digital libraries, but it lacks the semantic metadata to provide reasoning and inference functions. These functions are necessary for the computer-interpretable descriptions, which are critical in the area of learning objects reusability, autoexec course generation, dynamic course decomposition, learning object mining, etc.

To this problem, a mapping from LOM to statements in an RDF model has been defined (Nilsson, 2003). However, RDF alone doesn't share some basic common structures that help to describe classes of learning objects and types of relationship between learning objects. Thus, we need more flexibilities and facilities for expressing meaning and semantics than what has in RDF. The Semantic Web provides a catalytic solution to this problem.

To enhance the knowledge representation of the XML-based markup language, the traditional Semantic Web approach is to upgrade the original XML-based to ontology-based markup language. The upgrade mentioned above from XML-based LOM to RDF-based LOM is an example. The main problem with this approach is that the original XML-based markup language is replaced with the new ontology-based markup language. In this paper, we propose a novel integration approach that combines the first four layers of Semantic Web stack, including URI layer, XML layer (LOM), ontology layer, and rule layer. In the multi-layered semantic framework, Semantic Web technologies can be integrated with LOM to enhance the computer reasoning, and the original LOM still exist to cooperate with ontologies and rules. That is, the multi-layered semantic framework does not change the original schema of LOM. Hence, the existing LOM and SCORM metadata documents can continue to be used.

In order to demonstrate the feasibility of this multi-layered semantic framework, an application system of the Semantic-Based Learning Objects Annotations Repository (SLOAR) is developed to dynamically provide the information of relevant learning objects for course creators. SLOAR supports three different approaches to finding relevant learning objects, including LOM-based metadata, ontology-based reasoning, and rule-based inference. Such dynamic finding is desirable for a number of reasons. Firstly, it is customized for each

Kao S. and Hsu I. (2006).
INTEGRATING SEMANTIC WEB REASONING INTO LEARNING OBJECT METADATA.
In *Proceedings of the Eighth International Conference on Enterprise Information Systems - SAIC*, pages 181-184
DOI: 10.5220/0002445101810184

individual learning object, based on what metadata and knowledge the learning object has shown so far. Secondly, because the content or category of a learning object may keep changing, dynamic finding provides more up-to-date suggestions than a static design. Thirdly, as the number of learning objects may be large, adding suggestion links may become cumbersome for the course developer. Lastly, it can also be used at run-time to help in the decision of what content model component to deliver to the learner.

This paper is organized as follows. The next section presents the multi-layered semantic framework. Section 3 gives the architecture of SLOAR. In Section 4, we illustrate how the SLOAR can employ Semantic Web technologies to provide different approaches for finding relevant learning objects.

2 MULTI-LAYERED SEMAMTIC FRAMEWORK FOR LOM

In this paper, we propose a novel integration approach to combine the first four layers of Semantic Web stack, including URI layer (learning objects), XML layer (Metadata, LOM), Ontology layer (OWL), and Rule layer (Rule Markup Language, RuleML), as shown in Figure 1.

The URI layer is composed of learning objects, which can be identified by URI. The XML layer is composed of LOM metadata that are XML-based metadata for describing learning objects. The Ontology layer provides OWL-based ontologies, which can enhance LOM to Semantic-based metadata, hence, improve reasoning capabilities of LOM. The Rule layer supports more complex inference than the Ontology layer, and builds RuleML rules on top of OWL ontologies. The ontology is based on description logics to provide sound and decidable reasoning. In contrast, the rule is a logic program, which can complement ontology to support more complex rule-based inferences.

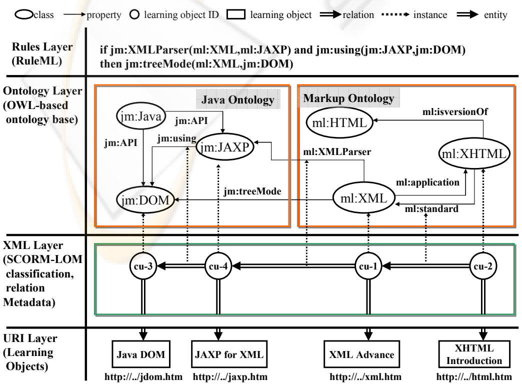

Figure 1: Multi-layered semantic framework.

3 SYSTEM ARCHITECTURE

The basic function of SLOAR is to provide the information of relevant learning objects for course creators. It supports three different approaches for finding relevant learning objects, including LOM-based metadata, ontology-based reasoning, and rule-based inference. In SLOAR, each learning object is associated with a classification metadata to quote extra semantic from a specific ontology class and is associated with a relation metadata to quote extra semantic from a specific ontology property. These ontologies are implemented in OWL that can be integrated into LOM, and as a result, the semantic capabilities of LOM were greatly improved.

The core components of SLOAR include the annotation base, knowledge base, search agent, and inference agent. The flow-oriented SLOAR architecture is depicted in Figure 2, as described in the following:

- **Annotations base:** is a learning object annotations repository that is composed of LOMs. A LOM is an XML document containing a set of markup elements to describe the learning objects..
- **Knowledge base:** is developed by the Semantic Web standard to support reasoning tasks. The knowledge is grouped into two categories: ontology layer inference using OWL-based ontologies and rules layer inference using RuleML logic program.
- **Search agent:** is a search engine that supports for a XPath query on the learning objects metadata base.
- **Inference agent:** is an intelligent agent that is implemented based on a JESS-based rule engine (JESS, 2005) and supports a XSLT processor.

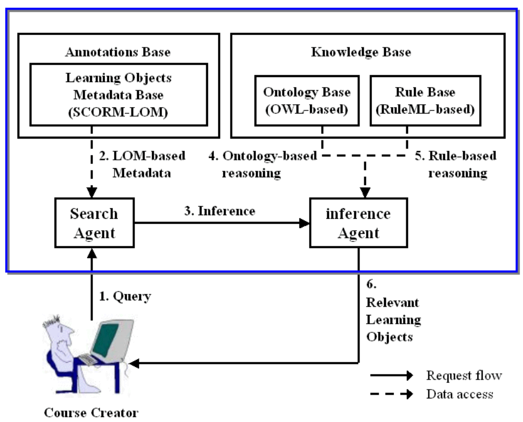

Figure 2: SLOAR Architecture.

The information flow of the SLOAR, as listed in the Figure 2, can be realized and started as follows.

1. The requester sends a learning object with URL to the search engine.
2. This step is the LOM-based metadata approach. The search agent relies on the request to query the LOM Base to finding all relevant LOM-

based metadata documents of the learning object.

3. The search engine sends these LOM-based metadata documents to invoke the inference agent.
4. This step is the ontology-based reasoning approach. The inference agent conducts the following tasks to infer for semantic links.

4.1 It retrieves and parses the relevant OWL-based ontologies quoted by the classification and relation tags.

4.2 It utilizes the OWL2Jess.xsl XSLT style sheet (OWL2Jess, 2005) for transforming these semantics of OWL-based ontologies into JESS-based rules.

4.3 It infers the semantic relevant learning objects from these JESS-based rules.

5. The rule-based inference is performed. The inference agent conducts the following tasks to infer the relevant learning objects.

5.1 It relies on the relevant ontologies, mentioned on the step 4.2, to query the rule base to retrieve relevant RuleML-based rules.

5.2 It utilizes the RuleMLTransform.xsl XSLT style sheet (RuleML2Jess, 2002) for transforming these RuleML-based rules into JESS-based rules.

5.3 It infers the rule relevant learning objects from these JESS-based rules.

6. Finally, the inference agent passes the information of relevant learning objects, including LOM-based, ontology-based, and rule-based learning objects to the requester.

4 USAGE SCENARIO OF SLOAR

To explicitly demonstrate how SLOAR works, a usage scenario of locating relevant learning objects is presented in the following.

4.1 Multiple-Layered Conceptions

Figure 1 provides a concrete example of how multiple-layered semantic framework is employed by SLOAR. In the URI layer, there are a number of learning objects, including XML Advance, XHTML Introduction, JAXP for XML, Java DOM, etc. In the XML layer, each learning object is described with a LOM that consists of classification metadata and relation metadata.

The user interface of SLOAR is shown in the Figure 3. A course creator selects a learning object and then presses the "Query" button. The SLOAR will rely on the learning object to invoke search agent and inference agent to produce information of relevant learning objects. In the next section, we will base on this example to explain SLOAR how to support different approaches to finding relevant learning objects.

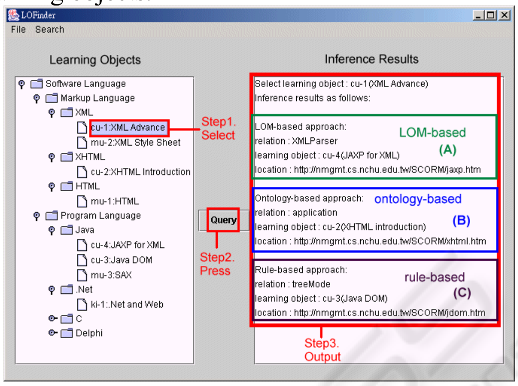

Figure 3: A query result on SLOAR with course ID is cu-1.

4.2 Different Approaches

When the inference agent receives LOMs from the search agent, it sequentially invokes different approaches for locating relevant learning objects.

4.2.1 LOM-based Metadata

According to the learning object ID (i.e. cu-1) received, the search engine finds all relevant LOMs in the annotations base. Since all LOMs are actually XML documents, this corresponds to performing an XPath query on each LOM, looking for learning object whose identifier has the same value as "cu-1". The search results consist of two LOMs. The former is the cu-1's LOM that consists of an outbound link from cu-1 to cu-4, as shown in Figure 4. The latter is the cu-2's LOM that consists of an outbound link from cu-2 to cu-1. The LOM-based approach only depends on the cu-1's LOM that exhibits a number of metadata. First, the *file* element describes the URL address of cu-1. Second, the *classification* element is used to describe where cu-1 quotes a particular ontology class. Third, the *relation* element is used to describe features that define the relationship between cu-1 and other learning objects. For example, the *kind* element describes where the relationship quotes a particular ontology property, and the *resource* element describes where cu-1 links to a particular learning object.

This kind of approach is that directly extracts data from the original LOM to produce the relevant information of learning objects, so we call it LOM-based metadata approach. The inference agent extracts data from the relation metadata of cu-1 to show that there is an *XMLParser* relation from cu-1 to cu-4. The output result of ontology-based reasoning is shown in (A) of Figure 3.

```
<resource identifier="cu-1">
<file href="http://…../SCORM/xml.htm"/>
<metadata><lom>……………….
<classification>
<purpose><source>URI</source>
<value>http://…/markup.owl#XML</value></purpose>
</classification>
<relation><kind><source>URI</source>
<value>http://…/java.owl#XMLParser</value></kind>
<resource><identifier>
 <catalog>learning object ID</catalog>
 <entry>cu-4</entry>
</resource></identifier></relation> ……………
</lom></metadata></resource>
```

Figure 4: The partial LOM code of the learning object cu-1.

4.2.2 Ontology-based Reasoning

The inference agent depends on semantics of Markup ontology and cu-2's LOM to reason the following facts.

1. The cu-2 learning object is an instance of *XHTML* class.
2. There is a *standard* relation from cu-2 to cu-1.
3. The *application* property is an inverse of standard property (see Figure 1).

Base on the above facts, inference agent can reason that there is an *application* relation from cu-1 to cu-2. The output result of ontology-based reasoning is shown in (B) of Figure 3.

4.2.3 Rule-based Inference

This inference agent relies on the previous inference results, LOM-based metadata documents and RuleML rules to perform the following tasks.

1. It converts the LOMs to JESS-based facts, as shown in Figure 5.

```
(assert (triple (predicate "http://../markup.owl#standard")
(subject "cu-2") (object "cu-1")))
(assert (triple (predicate "http:/./markup.owl#XMLParser")
(subject "cu-1") (object "cu-4")))
(assert (triple (predicate "http://../java.owl#using")
(subject "cu-4") (object "cu-3")))
```

Figure 5: The JESS-based facts.

2. It converts the RuleML rule (see Figure 1) to JESS-based rule, as shown in Figure 6.

```
(defrule XMLparserMode
(triple (predicate "http://../DO/markup.owl#XMLParser")
(subject ?x) (object ?y))
(triple (predicate "http://…/DO/java.owl#using")
(subject ?y) (object ?z))
=>
(assert
(triple (predicate "http://…/DO/markup.owl#treeMode")
(subject ?x) (object ?z))))
```

Figure 6: The JESS-based rule.

3. It relies on the above JESS-based facts and rule to infer the rule-based learning objects. The inference can infer that there is a *treeMode* relation from cu-1 to cu-3. The output result of rule-based reasoning is shown in (C) of Figure 3.

5 CONCLUSION

In this paper, an intelligent SLOAR prototype system is implemented. SLOAR is developed based on multi-layered semantic framework, including URI layer (learning objects), XML layer (LOM), Ontology layer (OWL), and Rule layer (RuleML). This framework is embedded into the Semantic Web stack and does not change the original schema of LOM. It results in making LOM computer-interpretable and hence enables automatically relevant learning objects finding.

Novel Semantic Web solutions, integrated with different types of high-level ontology-based metadata and XML-base rules, can dynamically tailor the knowledge base to take into account the user preferences for personalization. Thus, one future work is to extend the accessibility of the SLOAR towards the personalization model for individual-dependent dynamic courses according to user preferences.

REFERENCES

IEEE LOM, 2002. Retrieved Sep. 18, 2005, from http://ltsc.ieee.org/wg12/

JESS, 2003. Retrieved Sep. 18, 2005, from http://herzberg.ca.sandia.gov/jess/

Nilsson, M., et al., 2003. IEEE LOM RDF binding, Retrieved Sep. 18, 2005, from http://kmr.nada.kth.se/el/ims/metadata.html

SCORM: Sharable Content Object Reference Model, 2004. Retrieved Sep. 18, 2005, from http://www.adlnet.org

OWL2Jess, 2005. Retrieved Sep. 18, 2005, from http://www.inf.fu-berlin.de/inst/ag-nbi/research/owltrans/OWL2Jess.xsl

RuleML2Jess, 2002. Retrieved Sep. 18, 2005, from http://www.ruleml.org/jess/RuleMLTransform.xsl.