Proceedings of the 1st WSEAS / IASME Int. Conf. on EDUCATIONAL TECHNOLOGIES, Tenerife, Canary Islands, Spain, December 16-18, 2005 (pp18-23)

RuleML-based Ontologies for Specifying Learning Objects

Yevgen Biletskiy
Department of Electrical and Computer Engineering
University of New Brunswick
P.O. Box 4400, 15 Dineen Dr., D41 Head Hall
Fredericton, NB, E3B 5A3, Canada

David Hirtle
Faculty of Computer Science
University of New Brunswick
P.O. Box 4400, 540 Windsor St., Gillin Hall E126
Fredericton, NB, E3B 5A3, Canada

Abstract: - An approach to building ontologies that assists in achieving interoperability among semantically heterogeneous learning objects and learners is presented. Current work focuses on building ontologies of learning objects, particularly course outlines. The ontology is implemented using the Rule Markup Language (RuleML), an XML-based standard. A RuleML representation allows the ontology to be flexible, extensible and platform-independent. The ontology can be reused and integrated with other ontologies, and is easily converted to other XML-based languages (such as RDF and OWL) and even incorporated with existing XML-based repositories (such as IEEE LOM and CanLOM) using XSL Transformations. A small implemented example of the application of XSL Transformations to build the RuleML-based ontology is presented as a proof of concept.

Key Words: - Distance Education, e-Learning, Knowledge Representation, RuleML, XSLT

1 Introduction

The presented work is related to applications of the context mediation approach for achieving interoperability between semantically heterogeneous learning objects and learners. This paper is a logical continuation of the preceding paper "Conceptualization of Specifying Learning Objects" [1]. The approach of context mediation is described in [2,3], and its application to achieving interoperability between learning objects and learners is presented in [4]. This approach uses the domain of a *Common Ontology* [3] as specification of knowledge about learning objects to be delivered to learners and knowledge domains, which these learning objects belong to [4]. This common ontology must explicitly specify the learning objects' basic ontologies, relationships between learning objects, and also relationships between concepts of learning objects' metadata and learners' profile models for further conversion of learning objects to the learner's context. So, integration of learning objects ontologies into the common ontology must also involve some conversion procedures. The work presented here describes methods of building learning object ontologies with the goal of building the common ontology to assist in achieving semantic interoperability between learning objects and learners.

The definitions, basic principles and criteria for the design of ontologies (Gruber 1995) as well as some existing methodologies for building ontologies are reviewed (Fernandez Lopez 1999), and the *Skeletal Methodology* by Uschold and King, which provides general guidelines for developing ontologies, is selected as the basic approach for the task of building a common ontology in the context mediation for achieving interoperability among learning objects and learners. According to this methodology, the process of building the common ontology assumes three steps (Uschold 1996): ontology capture, coding ontology using a formal language, and integrating existing ontologies.

Following the Skeletal Methodology, we define the process of building and integrating ontologies of learning objects as consisting of the following steps:

1. Preprocessing – definition of data sources (learning objects) and consumers (learners) participating in data exchange.
2. Extraction of structure and content of learning objects participating in data exchange. This step allows the building of a learning object's basic ontologies and for it to be integrated in the common ontology.
3. Extraction and/or generation of learning object's metadata (LOM).
4. Ontology representation and coding.
5. Integration of learning object's basic ontologies involving LOM and conversion procedures, and finding links of semantic interoperability among learning objects and learners.

The particular focus in this paper is ontology representation and coding. This work uses the results of preceding work [1], which was devoted to the extraction of learning object structure, content and metadata. Before discussing the developed methods, we first review some existing methods, technologies, languages and standards, which can be used during the process of building learning object ontologies.

2 Overview of Methods and Standards for Building Learning Object Ontologies

An Educational Modeling Language (EML) is a set of semantic rich notions to describe pedagogical entities. These entities could be objects, designs or activities. They are used to create highly-structured course

Proceedings of the 1st WSEAS / IASME Int. Conf. on EDUCATIONAL TECHNOLOGIES, Tenerife, Canary Islands, Spain, December 16-18, 2005 (pp18-23)

material. An EML-based course might offer features such as: re-useable course material, personalized interaction for individual students, media independence, etc.

SCORM (Sharable Content Object Reference Model) is a standard specifying learning objects, and these specifications allow reusing Web-based learning content among various environments and products. This standard, developed by Advanced Distributed Learning (ADL), provides inexpensive development and maintenance of learning objects and flexibility to adapt learning to particular situations.

IEEE LOM (Learning Object Metadata) provides structural descriptions of reusable learning objects. The learning object metadata should be represented in XML (the schema for which is provided by IEEE) and attached to the learning object. Although it is an approved IEEE standard supporting XML and RDF, the standard contains too much metadata (67 elements) considering that information vendors and users are responsible for its implementation. CanLOM (Canadian Learning Objects Metadata Repository) is a repository of metadata based on the CanCore recommendations, which have several implementations, but it is still not a standard.

XML Schemas define the structure, data and content of XML documents. They also help machines to carry out the semantic meaning of the rules set by humans. RDF (Resource Definition Framework) is designed for representing the metadata of resources on the web. The “statements” in RDF describe real world objects such as papers or persons, or web pages. The “resources” and “properties” are described with RDFS (RDF Schema). RDFS extends RDF by model objects like classes, class inheritance, property inheritance, and domain and/or range restriction. The OWL (Web Ontology Language) is a language designed for denoting web ontologies. The semantic meaning carried by the document is processed by applications using OWL, rather than by humans. Machine readable information from the document is the design goal of OWL. It can be used to explicitly represent the meaning of terms in vocabularies and the relationships between those terms.

The Rule Markup Language (RuleML) is an XML-based markup language for publishing and sharing rules used for derivation, query, transformation, integrity checking and reactive behavior. In today’s World Wide Web, rules play an important role in the Semantic Web and Web Services [5]. They are being used in many application domains such as Engineering, e-Business, Law, and Artificial Intelligence. RuleML consists of a hierarchy of sublanguages to maximize interoperability by accommodating related technologies such as RDF and OWL. RuleML is the most appropriate knowledge representation for this work because it is a neutral interchange format allowing the representation of both concepts and the relationships between them in one document.

3 Approach to Building Learning Object Ontologies

As mentioned in the preceding paper [1], there are two general classes of learning objects: specifying learning objects, which specify the use of other learning objects, and resources, which store actual learning content. Because the specifying learning objects partially specify the referenced resource’s metadata, we propose decomposition of specifying learning objects (i.e. course outlines) and then building their ontologies. A typical course outline document contains the following four important parts:

1. Document metadata (author, location, language, etc.).
2. Course context (description, credit, instructor, institution, etc.).
3. Course structure (schedule of topics).
4. Recommended resources.

So, the ontology of a course outline can be represented as in figure 1, where {T} is a sequence of topics and {R} is a set of referenced resources.

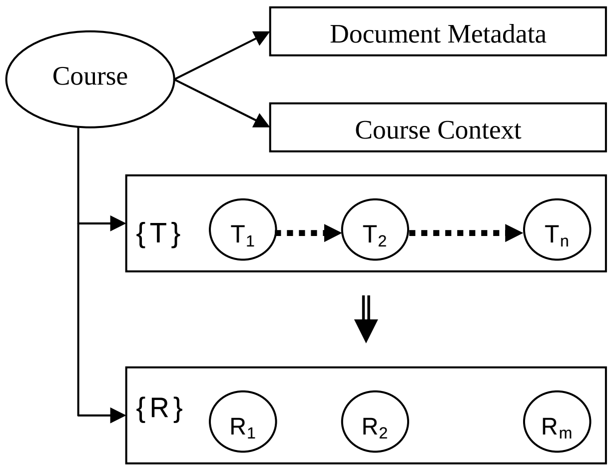

Figure 1 – Ontology of a course outline

The document’s metadata as well as course context partially specify the resource’s metadata and relate the resources to the course topics, therefore the task of capturing and coding the course outline’s ontology is important for further delivery of learning objects to learner’s context.

Typical course outline documents are stored in formats that can easily be converted to HTML. The preceding paper described a method of data extraction from semi-structured HTML representations of course outlines generated by word-processing tools such as MS Word, WordPerfect, Adobe Acrobat, etc., and conversion into a meaningful XML representation. The presented work is devoted to transformation of the meaningful XML representation to a RuleML-based ontology using XSLT.

3 Conversion of Meaningful XML into RuleML

3.1 XSL Transformation for Converting Meaningful XML Representation of Learning Objects into RuleML-based Ontology

The Extensible Stylesheet Language (XSL) is a family of recommendations by the World Wide Web Consortium (W3C). At the heart of XSL is the stylesheet, which specifies how a given XML document (or a class of such documents) is to be presented. The goal of XSL, as with Cascading Style Sheets (CSS) and HTML, is the separation of pure content (in XML format) from presentational considerations (contained in a stylesheet). To be displayed, the XML document must be converted via what is known as XSL Transformations (XSLT) [6].

The flexibility of XSLT as a language for transforming XML naturally lead to its use for general XML processing. In fact, it has many applications outside of the original XSL framework for which it was designed: generating web pages, sorting data, and even translating between XML and/or XML-based languages.

As with natural language, the need for translation between XML-based languages is great. The field of e-Learning is no different. The vital step of combining metadata and learning objects can only be successful on a large scale with a consistent metadata framework and accepted indexing guidelines. Unfortunately, the current XML-based standard, IEEE LOM, is not viewed as satisfactory by all interested parties. As a result, numerous organizations worldwide have developed their own specifications: CanCore, Dublin Core, uklomcore, etc. Though such variation hampers the usefulness of LOM, interoperability is still possible by translation among the diverse specifications. Translation between all the various specifications would require many translators, but an intermediate format can drastically reduce this number. RuleML, a shared rule markup language, is particularly well suited for the task not only because of its flexibility, but also because of the advantages accompanying representation within a rule base:

1. Support for weights, facilitating agent matching.
2. Access to various tools such as the Weighted-Tree Similarity Algorithm [7].
3. Derivations, transformations, and other general rule-related operations are possible in the same formalism.

As a neutral interchange format, RuleML also facilitates the translation of this metadata to (and between) various other Semantic Web representations such as RDF, OWL and SWRL [8]. A translation between CanCore and RuleML is already in place [9] and RuleML and RDF interoperability is already widely discussed in the community [10].

The (abbreviated) sample illustrated in Appendix A1 demonstrates this mapping between the meaningful XML representation of an LO and RuleML, as realized by an XSLT stylesheet (fig.2). The full RuleML representation output by the stylesheet is given in Appendix A3.

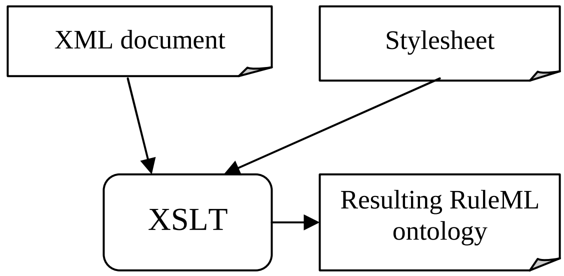

Figure 2 – Level 0 Data Flow diagram of the XSLT

3.2 Further Use of RuleML-based Ontologies in Context Mediation

As mentioned above, the RuleML representation allows including rules that formalize conversion of learning objects to the learner's context. The resulting RuleML document (Appendix A3) contains the same four sections as the represented course outline. The sections are separated by the following elements:

```
<Ctor>properties</Ctor>
…
<Ctor>structure</Ctor>
…
<Ctor>courseResources</Ctor>
…
<Ctor>courseSchedule</Ctor>
```

The document can be enriched by extra context attributes and easily integrated with RuleML representations of other learning objects, learner's contexts and conversion rules. Thus, the common ontology, which involves knowledge about learning objects and context conversions, can be built. The context conversions (or conversion procedures, for example language translator) can be applied to learning according to conversion rules, which connect the contexts (C) of a learning object (LO) to be delivered to a learner (L) (fig.3).

Conclusion

The presented work describes an approach to building learning object (particularly course outline) ontologies using the Rule Markup Language. Using an XML representation of learning objects as input, the XSLT stylesheet transforms it to RuleML, which is flexible, extensible, platform-independent and easy to parse. Recommendations to use ontologies in the context mediation approach for achieving interoperability between semantically heterogeneous learning objects and learners are also presented. Course descriptions (outlines) belong to the class of learning objects that specify other learning objects. It is important to find

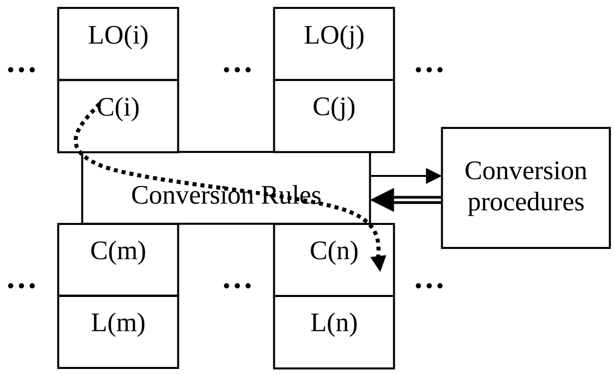

Figure 3 – Use of the RuleML-based Ontology

interoperability between specifying learning objects because such learning objects reference other, semantically different resources. The references from specifying learning objects are helpful, but application of the specifying learning object's metadata to referenced resources is not always comprehensive and is a challenging opportunity for further research.

References:

[1] Yevgen Biletskiy, Tim Scribner. "Conversion of Learning Objects to Meaningful XML". *The 8-th IASTED International Conference on Communication, Internet and Information Technology*, Boston, USA, October 2005.

[2] Firat A., Madnick St., Grosof B. (2002) Financial Information Integration in the Presence of Equational Ontological Conflicts, *MIT Sloan Working Paper No. 4408-02; CISL Working Paper No. 2002-16.*

[3] Goh, C. H., Madnick, S. E., Siegel, M. D. (1994) Context interchange: overcoming the challenges of large scale interoperable database systems in a dynamic environment, *In Proceedings of the Third International Conference on Information and Knowledge Management,* Gaithersburg, Maryland, USA, 337-346.

[4] Biletskiy Y., Vorochek O., Medovoy A. (2004) Building Ontologies for Interoperability Among Learning Objects and Learners, *The 17th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE)*, Ottawa, Canada. *Lecture Notes in Computer Science, Springer-Verlag,* vol. 3029/2004, pp. 977-986.

[5] David Hirtle, Harold Boley, Benjamin Grosof, Michael Kifer, Michael Sintek, Said Tabet, Gerd Wagner, "The Rule Markup Initiative", *http://www.ruleml.org*

[6] J. Clark. XSL Transformations (XSLT) Version 1.0. Recommendation REC-xslt-19991116, W3C, November 1999.

[7] V. C. Bhavsar, H. Boley, and L. Yang. A Weighted-Tree Similarity Algorithm for Multi-Agent Systems in e-Business Environments. *In Proc. Business Agents and the Semantic Web (BASeWEB) Workshop*, pages 5372. NRC 45836, June 2003.

[8] Ian Horrocks, Peter F. Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosof,Mike Dean, *"SWRL: A Semantic Web Rule Language Combining OWL and RuleML,"* W3C Member Submission 21 May 2004.

[9] Hirtle, D. and Z. Sun, 2003, "CanCore ⇔ WOO RuleML", Internal Report, Faculty of Computer Science, University of New Brunswick, www.cs.unb.ca/agentmatcher/translators.

[10]Harold Boley, Jing Mei, Michael Sintek, Gerd Wagner, "RDF/RuleML Interoperability", W3C Workshop on Rule Languages for Interoperability, Position Paper: 27-28 April 2005.

Proceedings of the 1st WSEAS / IASME Int. Conf. on EDUCATIONAL TECHNOLOGIES, Tenerife, Canary Islands, Spain, December 16-18, 2005 (pp18-23)

Appendices

Appendix A1. Example of Mapping between Meaningful XML and RuleML

XML

```
<document>

  <properties>

    <title>
     EE4283 – VLSI Systems
    </title>
    ...

    <language>
     EN-GB'
    </language>
  </properties>

  <structure>

    <course>

      <courseTitle>
       EE4283
      </courseTitle>
      ...
    </course>
  </structure>
</document>
```

RuleML

```
<Assert xmlns="http://www.ruleml.org/0.89/xsd">
  <And mapClosure="universal">
    <Atom>
      <Rel>document</Rel>
      <Cterm>
        <Ctor>properties</Ctor></opc>
        <slot>
          <Ind>title</Ind>
          <Data>EE4283 – VLSI Systems</Data>
        </slot>
        ...
        <slot>
          <Ind>language</Ind>
          <Data>EN-GB'</Data>
        </slot>
      </Cterm>
      <Cterm>
        <Ctor>structure</Ctor>
        <Cterm>
          <Ctor>course</Ctor>
          <slot>
            <Ind>courseTitle</Ind>
            <Data>EE4283</Data>
          </slot>
          ...
        </Cterm>
      </Cterm>
    </Atom>
  </And>
</Assert>
```

Appendix A2. XSLT Stylesheet for Converting Meaningful XML into RuleML

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns="http://www.ruleml.org/0.89/xsd"><xsl:output method="xml" indent="yes"/>

  <!-- Matches the root element, making it the relation of the whole atom -->
  <xsl:template match="/*[1]">
    <Assert xmlns="http://www.ruleml.org/0.89/xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xsi:schemaLocation="http://www.ruleml.org/0.89/xsd http://www.ruleml.org/0.89/xsd/hornlog.xsd">
      <And mapClosure="universal">
        <Atom>
          <Rel><xsl:value-of select="name(.)"/></Rel>
          <xsl:apply-templates select="*"/>
        </Atom>
      </And>
    </Assert>
  </xsl:template>
  <!-- Matches all elements (except root), creating Cterms for any with children and slots otherwise -->
  <xsl:template match="*|@*">
    <xsl:choose>
      <xsl:when test="count(*) &gt; 0">
        <Cterm>
          <Ctor><xsl:value-of select="name(.)"/></Ctor>
          <xsl:apply-templates select="*"/>
        </Cterm>
      </xsl:when>
      <xsl:otherwise>
        <slot>
          <Ind><xsl:value-of select="name(.)"/></Ind>
          <Data><xsl:value-of select="."/></Data> <!-- Could add XSD built-in datatype -->
        </slot>
      </xsl:otherwise>
    </xsl:choose>
```

```
  </xsl:template>
</xsl:stylesheet>
```

Appendix A3. Example Learning Object after Translation into RuleML by XSLT

```
<?xml version="1.0"?>

<Assert
xmlns=
"http://www.ruleml.org/0.89/xsd"
xmlns:xsi=
"http://www.w3.org/2001/XMLSchema
-instance"
xsi:schemaLocation=
"http://www.ruleml.org/0.89/xsd
http://www.ruleml.org/0.89/xsd/
hornlog.xsd">
 <And mapClosure="universal">
  <Atom>
   <Rel>document</Rel>
   <Cterm>
    <Ctor>properties</Ctor>
    <slot>
     <Ind>title</Ind>
     <Data>EE6213</Data>
    </slot>
    <slot>
     <Ind>author</Ind>
     <Data>Yevgen
     Biletskiy</Data>
    </slot>
    <slot>
     <Ind>lastAuthor</Ind>
     <Data></Data>
    </slot>
    <slot>
     <Ind>created</Ind>
     <Data>2005-01-
     05T13:12:00Z</Data>
    </slot>
    <slot>
     <Ind>pages</Ind>
     <Data>1</Data>
    </slot>
    <slot>
     <Ind>words</Ind>
     <Data>154</Data>
    </slot>
    <slot>
     <Ind>characters</Ind>
     <Data>882</Data>
    </slot>
    <slot>
     <Ind>company</Ind>
     <Data>UNB</Data>
    </slot>
    <slot>
     <Ind>version</Ind>
     <Data>10.4219</Data>
    </slot>
    <slot>
     <Ind>language</Ind>
     <Data>EN-GB'</Data>
    </slot>
   </Cterm>
   <Cterm>
    <Ctor>structure</Ctor>
    <Cterm>
     <Ctor>course</Ctor>
     <Cterm>
      <Ctor>courseName</Ctor>
      <slot>
       <Ind>courseCode</Ind>
       <Data></Data>
      </slot>
      <slot>
       <Ind>courseTitle</Ind>
      <Data>EE6213</Data>
     </slot>
    </Cterm>
    <Cterm>
     <Ctor>courseCredit</Ctor>
     <slot>
      <Ind>ch</Ind>
      <Data></Data>
     </slot>
     <slot>
      <Ind>wh</Ind>
      <Data></Data>
     </slot>
    </Cterm>
    <Cterm>
     <Ctor>courseTeacher</Ctor>
     <slot>
      <Ind>teacherName</Ind>
      <Data>Yevgen Biletskiy</Data>
     </slot>
     <slot>
      <Ind>teacherLocation</Ind>
      <Data>GWC115</Data>
     </slot>
     <Cterm>
      <Ctor>teacherContacts</Ctor>
      <slot>
       <Ind>teacherEmail</Ind>
       <Data>biletski@unb.ca</Data>
      </slot>
      <slot>
       <Ind>teacherPhone</Ind>
       <Data>447-3495</Data>
      </slot>
      <slot>
       <Ind>teacherUrl</Ind>
       <Data></Data>
      </slot>
     </Cterm>
    </Cterm>
    <Cterm>
     <Ctor>courseResources</Ctor>
     <Cterm>
      <Ctor>courseTextbook</Ctor>
      <slot>
       <Ind>textAuthor</Ind>
       <Data></Data>
      </slot>
      <slot>
       <Ind>textTitle</Ind>
       <Data></Data>
      </slot>
     </Cterm>
     <slot>
      <Ind>courseUrl</Ind>
      <Data></Data>
     </slot>
     <slot>
      <Ind>coursePrerequisite</Ind>
      <Data>EE 3232.</Data>
     </slot>
     <slot>
      <Ind>courseDescription</Ind>
      <Data>Tools and methods for the
      design of CMOS digital
      Application Specific Integrated
      Circuits.</Data>
     </slot>
    </Cterm>
    <Cterm>
     <Ctor>courseLecture</Ctor>
     <slot>
          <Ind>lectureLocation</Ind>
          <Data></Data>
         </slot>
         <slot>
          <Ind>lectureTime</Ind>
          <Data></Data>
         </slot>
        </Cterm>
        <Cterm>
         <Ctor>courseLab</Ctor>
         <slot>
          <Ind>labLocation</Ind>
          <Data></Data>
         </slot>
         <slot>
          <Ind>labTime</Ind>
          <Data></Data>
         </slot>
        </Cterm>
        <Cterm>
         <Ctor>courseSchedule</Ctor>
         <Cterm>
          <Ctor>scheduleTopic</Ctor>
          <slot>
           <Ind>topicWeek</Ind>
           <Data></Data>
          </slot>
          <slot>
           <Ind>topicName</Ind>
           <Data></Data>
          </slot>
          <slot>
           <Ind>topicText</Ind>
           <Data></Data>
          </slot>
         </Cterm>
        </Cterm>
       </Cterm>
      </Cterm>
     </Atom>
    </And>
   </Assert>
```