Ontological Support for Web Courseware Authoring System Based on SCORM

Jing Cheng

1. Faculty of Computer and Information Science, Southwest University, Chongqing, China, 400715, Email: cjcat@swu.edu.cn

Abstract

In the paper, we use ontology technology to construct a web courseware authoring system and realize the semantic interoperation in heterogeneous e-learning systems. We set up a domain ontology model which is the core component of the system, and can be related with the knowledge concept in domain ontology base with learning resources in the learning resources repository. Then we use the SCORM simple sequencing model to set up the clusters of learning activity to form the agile and adaptive course structure, and map the learning activity clusters encapsulated by SCORM onto OWL language. Based on the domain ontology, a semantic conflict elimination model is put forward, which is the foundation to resolve semantic confliction in authoring courseware cooperatively. Finally we prove that by using the technology of semantic web and all kinds of learning recourses standards we can resolve the interoperation of heterogeneous e-learning systems.

Keywords: Model Of Domain Ontology, The Sequence Of Learning Activities Based SCORM, Model Of Semantic Conflict Elimination

1. Introduction

Since 1990s’ distance learning has grown rapidly, and lots of web-based learning systems emerged. Because of the difference in architectures, communication mechanisms and data storage methods, these systems become more and more difficult for learning resources’ sharing, reusing, and exchanging. That means, interoperability in heterogeneous learning environments have been a crucial problem, and must be considered when designing the web learning system [1][2][3]. Simultaneously, the educational technology standards are moving forward at rapid pace, which include the IEEE LOM, Dublin Core, SCORM etc. These educational standards bring important contributions with respect to management of educational resources. Nowadays the most famous standards would be the ADL SCORM[4], which incorporates many specifications of the IMS[5]and the IEEE LOM[6], and be used to address semantic annotations, content aggregation and sequencing. However, SCORM has chosen its own XML formats and methodologies, thereby making it much more difficult in heterogeneous e-learning systems. So Semantic interoperability in heterogeneous e-learning systems is still an unsolved problem, and has got more and more attention.

To resolve these problems ontology technology are introduced in the fields[7][8]. Ontology is an advanced technology in knowledge engineering, and often used to construct the knowledge-based intelligent systems[9][10]. It provides the sharing definitions of basic terms and their relations in the task domain, and also provides axioms as rules and constraints for manipulating and managing these terms and relations within the common domain vocabulary. By the rules or constraints ontology can do some reasoning in knowledge base and have the capability to verify the knowledge consistency.

Nowadays ontology technology has been exploited to integrate the heterogeneous hypermedia-learning systems. For example, Lora Aroyo etc. have developed AIMS[11] and OntoAIMS[12], both of which typically include: ontology-driven subject domain, repository of learning resources, adaptive and personalized modules based on the conceptual domain knowledge. The difference between these two systems is that AIMS uses topic maps to implement the domain model, while OntoAIMS uses OWL and DAML-S. Another example is the Elena project[13][14], which defines a distributed, dynamic and heterogeneous educational environments based on peer-to-peer framework. In this context, Peter Dolog etc. developed the Personal Learning Assistant, which uses ontology to realize the personalized learner’s model supported the semantic interoperability in distributed and heterogeneous educational environments[15][16].
In this paper we are devoted to construct a web-based courseware authoring system. The courseware from our system is decomposed into independent modules, which can be combined to form new courses and reused in other system. When building the courseware authoring system, we set up the ontology-based subject domain model for sharing and reusing knowledge, and we use ontology technology and OWL language to realize the SCORM standard. In order to cooperative authoring, we make use of ontology to eliminate the semantic conflicts from the several courseware authors.

The framework of the paper is that: in the first section we introduced the ontology’s application in heterogeneous learning systems. In the second section we present the architecture of our system. In the third section we analysis the requirements of domain modeling in hypermedia learning systems and construct the domain model based on ontology technology. In the fourth section we discuss how to construct the course structure and map the sequence of learning activities onto OWL. In the fifth sections we discuss how to resolve the problem of semantic conflicts in cooperative authoring and provide the elimination model of semantic conflicts. Finally we present the prototype of our system and summary the whole paper.

2. Architectural description of the system

The system is designed to support the author of web-based courseware in construction, description and conceptualization of all kinds of educational materials and composition of the course structure. In order to facilitate the creation, editing, maintenance and reusability of teaching material and various coursewares, we put forward a system architecture driven by ontology. The authoring environment consists of three main application modules: Domain Editor, Resources Editor, and Courseware Editor. These three modules correspond to the three distinctive system information bases: domain ontology base, resources repository with relevant metadata, and courseware information base. Accordingly the system needs three different authoring roles: Domain Author, Resource Author, and Course Author. Please see Figure 1.

The Domain Editor enables the domain author to construct the mapping structure of domain concept based on ontology technology. It provides facilities to add, delete and update domain terms and their relations. For each new term the author specifies its name and definition along with its classification in a simple hierarchy. Also the author can create several kinds of relations among domain terms or set up references between domain terms and learning objects in the resources repository. Domain ontology base is the core module of system, in which the model of domain ontology and the relevant instances are stored.

The Resources Editor enables the resource author to manage a collection of educational materials. It provides several simple options, such as searching, editing, adding, updating and deleting learning materials. The most important function is to encapsulate the original learning materials to the learning objects. This option will annotate the original learning materials with the key terms in domain
ontology and bind them with metadata standards SCORM to form small learning units, that is the SCO’s (Sharable Content Object), which will declare the topic, presentation formats and instructional purposes of the generated learning objects.

The Course Editor allows the author to define the structure of courseware and to organize all the course relevant materials. In our system there are four different granularity courseware units: web pages, tasks, learning activities and courses. Web pages are the minimal units without study goals, and tasks are the elementary units with only one explicit goal, several tasks with some control information can consist of a learning activity, which have a special learning goal, and several learning activities compose a course. The Course Editor provides the edit functions for various courseware units, such as adding, modifying, deleting, etc. In order to create the course structure the author should (1) select concepts from the domain model and assign them to a learning activity; (2) select a specific sequence of tasks to form a learning activity; (3) assign course activities for one topic; and (4) link educational resources to each course topic.

3. Domain ontology model

Domain ontology model is the core of our system, which can control the relationships among domain concepts, and the mutual references within domain ontology base, resource repository and courseware information base. And the model can be instantiate by the ontology language. In this section we will present the domain ontology model, and then will instantiate the model by OWL ontology language[17].

In our authoring systems there are three important objects including knowledge concepts, learning resources and courseware, which are from the three layer of system information base respectively. According to the relations of these three objects, we construct the domain ontology model, see Figure 2.

In this model we use the rectangle to present classes and the ellipse to present properties, and then connect them with arcs. We use class Concept to annotate concepts, and class LearningObject to annotate learning materials in learning resources repository. Learning materials are used to describe concepts, so LearningObject is related to Concept by dc:subject property, and organized by dcterms:requires property. Concept uses subConceptOf property to connect the sub-concepts, and uses isPrerequisiteFor property to relate the prior concepts. And concepts often play a certain role in a given course document, including the introductive role, full descriptive role etc., so we present these facts with the instances of ConceptRole class and its two properties: isPlayedIn and isPlayedBy. We use class CoursewareDocument to present the course documents, which have several types, for example,
the most general document types are Educational Material and Examination Material, and each has the subtypes. According to these facts we set up the class DocumentType and two relevant properties: hasDocumentType and subTypeOf.

To illustrate the domain model above, we offer an example described by OWL, which is an ontology language and can support applications of reasoning and transformation. Following example shows how a courseware page can be annotated based on our domain ontology:

<table>
<thead>
<tr>
<th>Table 1. The example of domain ontology based on OWL</th>
</tr>
</thead>
<tbody>
<tr>
<td>sun_java: ‘ java/concepts/class.html ’</td>
</tr>
<tr>
<td>owl: type -&gt; doc:CoursewareDocument;</td>
</tr>
<tr>
<td>doc: OO_Class [</td>
</tr>
<tr>
<td>owl: type -&gt; doc: Concept;</td>
</tr>
<tr>
<td>doc: isPrerequisiteFor -&gt; doc: OO_Inheritance;</td>
</tr>
<tr>
<td>doc: subConceptOf -&gt; doc: Object-Oriented ] .</td>
</tr>
<tr>
<td>doc: ClassesIntroduction [</td>
</tr>
<tr>
<td>owl: type -&gt; doc: ConceptRole;</td>
</tr>
<tr>
<td>doc: isPlayedBy -&gt; doc: OO_Class;</td>
</tr>
<tr>
<td>doc: isPlayedIn -&gt; sun_java: ‘ java/concepts/class.html ’ ;</td>
</tr>
<tr>
<td>] .</td>
</tr>
<tr>
<td>doc: Introduction [</td>
</tr>
<tr>
<td>owl: Type -&gt; doc: DocumentType;</td>
</tr>
<tr>
<td>doc: subTypeOf -&gt; doc: Cover ] .</td>
</tr>
</tbody>
</table>

4. Composition of course structure by SCORM

A course structure is defined as a sequence of course component (Learning Activities) interconnected in order to achieve some special learning goals. In this section we will introduce how to set up the course structure by SCORM and how to map them onto ontology language OWL.

4.1 Sharable content object reference model (SCORM)

The SCORM or Sharable Content Object Reference model is a collection of existing or newly developed specifications driven by the Advanced Distributed Learning (ADL) Initiative. The goal is to establish a new distributed learning environment that permits the interoperability of learning tools and course content on a global scale. The main parts of the SCORM specification are the SCORM content packaging model which mainly borrowed from the IEEE-LOM, the SCORM simple sequencing model and the SCORM specific runtime. The SCORM model assumes that content is built up out of small learning units, the SCO (Sharable Content Objects). A SCO is intended to be a self-contained unit of learning. Courses are designed by creating an packaging manifest document named as organizations. An organization is comparable to a document tree structure, whose leaves consist of SCO’s and whose nodes(Items) represent the course internal structure. Each structuring level (manifest, organization, item and SCO) can be annotated with metadata. The separation of content and organization means that the same SCO can be reused in different organization documents.

The SCORM simple sequencing model provides the model of learning activities, which can be organized as tree of clusters of more primitive activities to form the course structure. So we will introduce the definition of learning activity firstly, and then we will give an illustration how the primitive activities are organized as the tree structure of a course.

4.2 Learning activities

The learning activities (Figure. 3) capture learning-specific tasks within the context of a selected course. Each primitive learning activity LA is determined by a tuple:

<Activity_ID, Input, Output, PreCond, Effect, Composite_Components>.
Thus it specifies some prerequisites (pre-conditions) and some input and produces some effect (post-conditions) and some output, which is further used for the selection of the following activity, and it may have an internal structure.

Chains of atomic and composite learning activities are realized by applying the same composition operators. Figure 4 describes the sequencing process corresponding to the course clustering structure.

4.3 Mapping the sequence of learning activities onto OWL

Here we give an example of how the SCORM-based course clustering in Figure 4 can be translated into OWL specification. We use the N3 notation for the specification with owl denoting the OWL, and scorm denoting the SCORM namespace. A partial binding of SCORM concepts expressed in an OWL ontology used in the example is also included.
Table 2. The example of Learning Activities expressed by OWL

```
:Course a scorm:LearningActivity;
  owl:consistsOf
    [owl:Sequence;
      owl:components
      (:Module0 :Module1 :Module2)
    ].

:Module1 a scorm:LearningActivity;
  owl:consistsOf
    [owl:Choice;
      owl:components
      (:Lesson01 :Lesson02)
    ].

:Module2 a scorm:LearningActivity;
  owl:consistsOf
    [owl:Unordered;
      owl:components
      (:Lesson01 :Lesson02 :Lesson03 :Lesson04)
    ].

:Module0 a scorm:LearningActivity
  owl:consistsOf
    [owl:If-Then-Else;
      owl:ifCondition
        {:Lesson02 scorm:objectiveMeasureGreater “0.75”};
      owl:then :Lesson01;
      owl:else :Lesson02
    ].
```

5. Elimination of semantic conflicts

Cooperative authoring is a very complex process, which needs dialogue support, asynchronous usage of authoring elements, such as course materials, ontology, and basic resources, etc. During this process, the misunderstanding about certain article segments between cooperative authors occurs frequently. The reason lies in the different languages, areas and educational background of cooperative authors. We called this as “Semantic Conflicts”. To resolve the problems of semantic misunderstanding, we advance the Elimination Model of Semantic Conflicts, which can provide the mechanisms to help cooperative authors to form the consistently semantic understanding about conflicting segments.

Firstly let us to analysis the whole process when semantic conflicts arise. In this process there are three correlative steps:

- To tag the misunderstanding segments by the interpellator who is the author to put forward questions,
- To annotate these tagged segments with the terms of domain ontology by interpreter, who may be the domain experts,
- And to send the results of interpretation to the other relevant cooperative authors.

See Figure 5.

![Figure 5. The process to elimination semantic conflicts](image)
Basing on the analysis above, we advance the Elimination Model of Semantic Conflicts, which is a five tuple \( M = <\text{Role}, \ k, \ \rho, \ \mu, \ \lambda> \), where

- **Role** is the set of actors to participate in the elimination of semantic conflicts, and **Role** = \{interpellator, interpreter, and embracer\}. Interpellator is the person who questions the terms or sentences causing semantic conflicts and tags the misunderstanding segments. Interpreter is the person to interpret these tagged fractions based on domain ontology and to associate these fractions with the terms in ontology. Embracer is the person to receive the associated ontology knowledge to continue the authoring.
- **k** represents the terms or sentences caused semantic conflicts.
- **\( \rho(k) \)** is the trigger option to tag the segments **k** and activate the interpreter to annotate **k**.
- **\( \mu(k) \)** is the interpretative option for the tagged segments **k**, which will interpret **k** to annotate the fractions with the terms of domain ontology.
- **\( \lambda(k) \)** is the option to send the results of **\( \mu(k) \)** to the relevant authors.

6. System prototype

We have implemented a system prototype based on the technologies the paper had discussed above, and the interface of our system have displayed as Figure 6. In the left of window there is a main edit area of all objects such as courseware, concepts of domain ontology. And as we have seen, an ontology instance described by RDF graph is showed in the left window, which has been transformed into the codes of OWL ontology language in the background. In the right of the window there are three tabs about project, domain ontology, and courseware respectively.

![Figure 6. The interface of prototype](image)

7. Conclusions

In the paper we discuss several fresh technologies to set up a courseware authoring system such as ontology technologies, cooperative authoring technologies, etc. Based on these technologies a system’s prototype is implemented. By setting up domain ontology model, we can structure the domain knowledge and make the subject knowledge more systematic. Then we use the SCORM simple sequencing model to set up the clusters of learning activity to form the course structure. By mapping the clusters of learning activity onto OWL ontology language we can make use of semantic web technologies to resolve semantic interoperability in heterogeneous learning systems. And by domain ontology model we can eliminate the semantic conflicts among different cooperative authors. So we argued that the adoption of semantic web technologies is useful for learning technologies to integrate heterogeneous infrastructure, tools and business processes in different educational environments.

8. References

[17] Web Ontology Language (OWL), http://www.w3.org/TR/owl-ref/