i-OntoLearning: Ontological support for rich learning

Minh Tan Vu[1] and Angela Goh[1],

[1] Nanyang Technological University,
Nanyang Avenue, Singapore
{y050005,asesgoh}@ntu.edu.sg

Abstract. Many university and college programs have a rich repository of learning resources. These resources could range from multi-media material to simple text documents, from powerpoint slides to web-based applets. However, they are created by different authors and are usually accessible by the students registered for the courses taught by these authors. In order to enable students to discover the relationships that exist amongst these resources, we have designed i-OntoLearning, a system based on the Computing Ontology [6] and a Metadata Ontology which we have created. This platform supports learning through the ability to infer relationships within the curricula as well as with the learning resources. As more resources are added to the system, further interesting relationships can be discovered. This paper focuses on the application of semantic web technologies including ontology design and queries on these ontologies.

Keywords: Computing Ontology, metadata ontology, learning resources

1 Introduction

Academic courses are often designed in a silo-like manner. Students frequently see little connection between courses, whether these connections are 'vertical' (freshman to senior years) or 'horizontal' (courses within the same year of study). This often leads to poor overall integration of knowledge and weak application of foundation courses in more advanced courses. For example, if a Computer Science senior is taking a course in cryptography and needs to design software that requires a hashing function, material covered in earlier courses such as Data Structures and Algorithms will be needed. However, students may only have a vague recollection of this knowledge and searching across different course sites is not easily facilitated. Furthermore, the fact that Hashing is required in several other modules throughout the program may not be evident to the student.

Lecturers who are teaching the courses may place various resources in different repositories. Over the years, a very rich store of such resources would have accumulated.

For example, in a typical Algorithms course, there could be images, software, slides, videos, documents and quizzes all linked to the course.

In order to encourage active learning by university students, we propose i-OntoLearning, a platform that will support a rich media repository, whose contents are organized using ontologies. In this paper, we will present the ontology construction and some sample queries. Computer Science has been selected as a domain but it should be noted that the approach adopted in i-OntoLearning can be adapted for any other discipline, such as chemistry, geography and medicine.

2 Related Work

There has been wide interest in recent years on the use of ontologies and reasoning to personalize e-learning [1]. The notion of integrating a learner's profile with metadata of learning resources has triggered much interest among education technologists. Saini and Ronchetti [2] derived an ontology based on the ACM Computing Curricula and used it to create navigation paths between e-learning resources that have related metadata. Resources are classified into topics in the ontology and two resources that have sibling topics will be related. However, properties between topics in the ontology are added subjectively by the author of the ontology and authors of resources do not have the control to add links between metadata and topics. The use of ontologies to model and retrieve multimedia e-learning resources has also been reported [3, 4].

Tan and Goh [5] proposed a framework for semantics-based classification, navigation and query using ontologies. Resources are first classified into specific domains using ontologies with metadata information. Resources are therefore associated with concepts in the ontologies facilitating navigation via a conceptual map. Using this map, user can view resources of related concepts before zooming into specific details. Finally, conceptual-based queries can be made to produce more relevant and precise result. However, the paper did not illustrate how the ideas could be realized.

This paper presents a realization of the above including semantics-based navigation and query approach, integration of multimedia resources and curriculum and metadata structures.

3 Ontology Construction

In this section, we discuss the structure of various ontologies used in i-OntoLearning. The design target is to build ontologies in both the metadata layer and the Content Structure Modeling layer. The Computing Ontology project [6] is used as a foundation upon which specific curricula offered by different universities can be modeled. In this work, we have adopted OWL [7] which is currently a de-facto standard for ontology description and which has also been used in the Computing Ontology. An ontology includes descriptions of classes, instances of classes, properties, as well as range and domain constraints on properties. It also contains various types of relationships between classes or properties. These relationships form the structure of the ontologies. The most common relationship is the sub or super class relationship which determines the hierarchical structure of an ontology.

The Computing Ontology facilitates the ability to discover implicit relationships between learning resources. Furthermore, we have designed a Metadata Ontology which records metadata information. This enables users to distinguish between different types of learning resources. The content structure of resources is modeled using the IMS Content Packaging Ontology which resembles IMS Content Packaging Information Model [8]. These three ontologies form the basis upon which university-specific course ontologies are built. Fig. 1 shows the relationship between these ontologies. It should be noted that two university-specific courses (csc202 and csc105) are used as examples. A typical university curriculum includes many other such courses but are left out of the figure in order to simplify the discussion. The following sections will elaborate on each of the ontologies.

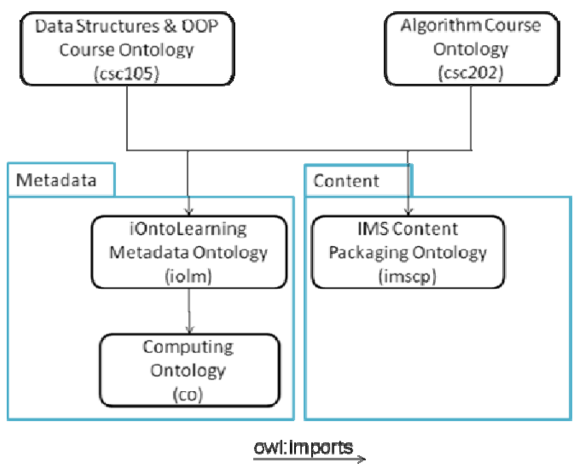

Fig. 1. i-OntoLearning Structure

3.1 The Computing Ontology

In 2005, with funding from the US National Science Foundation (NSF) and support from the ACM Education Board, the Computing Ontology was started [6]. The project intends to build an ontology covering the whole field with eight levels of details. The Computing Ontology is used as the basis of our Metadata Ontology. Implicit relationships between resources are inferred based on its reference to the Computing Ontology. If two resources refer to a topic in the Computing Ontology, they are deemed to be related to each other. However, the Computing Ontology is an on-going project. The current version contains mostly classes from level 5, 6, 7 and some from level 4 and 8. Only a small part of these classes contains instances. There is also a lack of relationships between classes. We have therefore modified the ontology. This extension is based on Curriculum 2005 [9]. For instance, advanced data structures are provided but fundamental data structures are not included. Examples of modifications are given below. These will be used in the queries in Section 4.

For Data Structures, we added a sub-topic called 'Fundamental Data Structures' and further divided it into two subclasses: 'Primitive Types' and 'Reference Types'. In 'Reference Types', we defined a subclass called 'Linked Structures', containing instances such as Linked Lists, Stacks, Queues, Graphs and Trees. For 'Object-Oriented Programming', we added a sub-topic for 'Language Design' called 'Object-Oriented Design' and a subclass for 'Language Features' called 'Object-Oriented Features'. These classes contain object-oriented concepts such as Information Hiding, Encapsulation, Inheritance, Polymorphism and so on.

For each class, we created an instance to represent it. This instance is used in properties with the Metadata Ontology which will be explained in the next section. When the Computing Ontology is completed, these instances will not be needed and actual instances will be used instead.

3.2 i-OntoLearning Metadata Ontology

Learning resources form a major part of i-OntoLearning. In each degree program, it is anticipated that there will be vast amounts of such resources. We therefore require metadata information to identify and describe each resource. Dublin Core [10] and Learning Object Metadata (LOM) [11] were explored as the basis of this ontology. Both support topic categorization. However, while Dublin Core is too simple, LOM contains information that is unnecessary in this system. For example, information regarding cost of developing the resources and other information were deemed unnecessary for this prototype. It was therefore decided that the Metadata Ontology be based on a small, best-practice set of properties.

The Metadata Ontology contains only one class called FileMetadata and eight data-type properties: *level, language, format, size, href, learningResourceType, title, and creator*. *Title, language, format, size, learningResourceType* are adopted from LOM [11]. *Href* contains the link to the resources. *Creator* indicates the person/entity that is responsible for the content of the resource. *Level* refers to the Aggregation Level in LOM, though we use a defined vocabulary instead of numbers. Learning objects can be classified into Course, Module, Lesson, and Component. It can be seen that the Metadata Ontology adopts a subset of LOM but has discarded the Program level since the ontology only models courses and lower level learning objects.

The Metadata Ontology contains several properties to utilize the Computing Ontology. We utilize three properties from the Computing Ontology, namely, *isA*, *uses* and *isPartOf*. Property *isA* is used when a topic is an instance of another topic, while *isPartOf* is used when a topic is a sub-topic. For example, “Minimum Spanning Trees” is an instance of “Trees” but “Tree Traversal” is a part of “Trees”. These three properties are used to model relationships between courses, whereas in the Metadata Ontology, they are used to relate resources with topics in the Computing Ontology. A resource has *isA*, *uses*, *isPartOf* relationships with topic when that resource has corresponding properties with the Computing Ontology element. One question that may arise is, what if the resource is related to an entire topic itself. To resolve this, we created an additional property, which is called *isClassifiedAs*. Furthermore, one might want to find out if there is a relationship between a specific resource and a topic. Property *relatedTo* is added as the super-property of the aforementioned properties.

3.3 IMS Content Packaging Ontology

I-OntoLearning’s metadata layer is made up of the Computing Ontology and Metadata Ontology. However, there is still a need to build the composition structure of resources, which is called Content Structure Modeling layer. The purpose is to clearly define the manner in which resources are packaged, aggregated and managed so that there is no ambiguity in interpreting what the resource is. Instructional Management System Content Packaging (IMS CP) [8] is chosen because of its flexibility and its interoperability with many e-learning systems. Another reason for selecting IMS is because the Singapore Standards for e-learning, SS 496 Parts 1 to 9, are based on this standard. It should be highlighted that the design of i-OntoLearning is such that this Content Packaging Ontology could be replaced by another standard such as Open Archives Initiative Object Reuse and Exchange (OAI-ORE) [12]

IMS CP consists of two main parts: resources and a special top-level XML file called Manifest which describes the organization of the content. An IMSCP Ontology was built to model this Manifest. Elements in the Manifest such as Organization, Item, Resources, File and Metadata are classes that have object properties among them or datatype properties for XML attributes. For instance, a Manifest may contain exactly one Resource and one Organization element. Therefore, the Manifest has object properties *hasOrganizations* and *hasResources* with cardinality of one. A Manifest contains attributes such as *identifier*, *xml:base* and *version*.

3.4 Specific Course Ontology

Specific Course Ontology contains individuals of the classes introduced in Metadata and IMSCP Ontology. For example, *FileMetadata* class in Metadata Ontology becomes a subclass of *Metadata* class in IMSCP Ontology.

In this paper, two Specific Course Ontologies are used to illustrate how i-OntoLearning supports queries for a rich learning experience. The two courses, "Data Structures and Object-Oriented Programming" and "Algorithms" highlight the flexibility of the system in supporting courses that are made up of multiple topics (such as data structures and programming languages). In fact, the two courses contain a total of 81 learning resources, most of which are at the module/chapter and lesson levels. The nature of these courses provides good examples of relationships between topics within each course.

4 Queries

In this section, we show how the New RacerPro Query Language (nRQL) is used to support queries in the prototype. RacerPro [13] was selected as the OWL reasoner due to its completeness and the fact that it can function as a separate server application. The current version supports requests from applications written in Java or Common Lisp. All queries presented below are implemented in a desktop client using RacerPro's Java client library. In order to demonstrate the i-OntoLearning prototype, a desktop client was created as shown in Fig. 2.

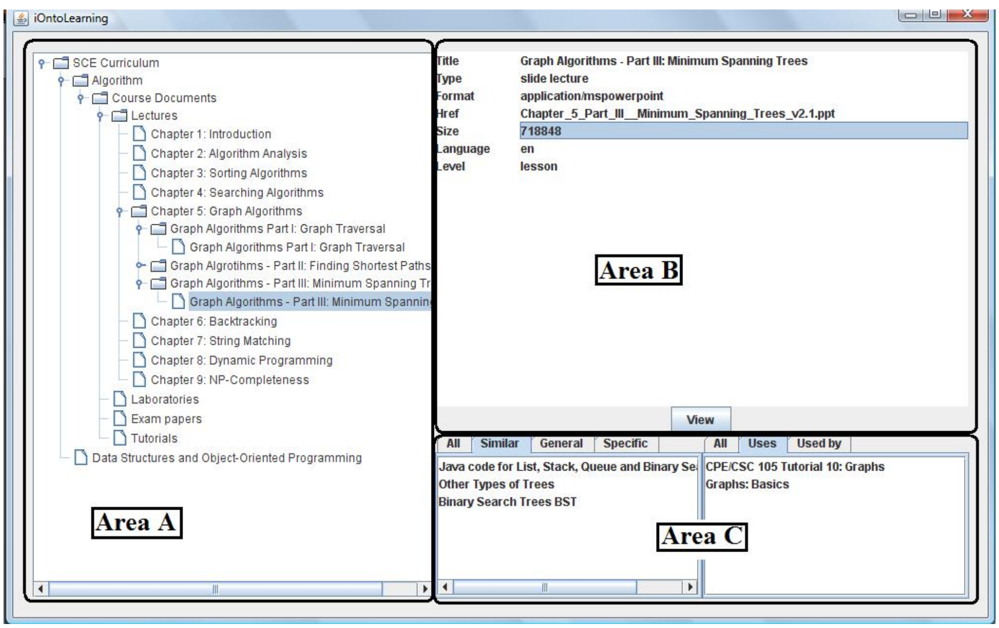

Fig. 2. Screenshot of i-OntoLearning's desktop client

The desktop client consists of three areas. Area A, called Content Navigator, provides the structure of courses based on i-OntoLearning ontologies. Area B, called Detailed Information Panel, shows metadata information of a resource and a button to open the file. Area C, called Related Resources Panel displays resources that have special relationships with the currently selected resource. Items in Area A and C are clickable, thus allowing learners to navigate through the hierarchical content structure (in Area A) or through the links between related items (in Area C). Information in Area B will be updated accordingly.

4.1 Content-related Queries

IMS CP Ontology provides the framework to organize resources into content structure. In IMS CP Ontology, an Organization element corresponds to a course. It contains a hierarchy of Item elements which define the structure of the course. In this section, we present the queries that facilitate navigating this hierarchy.

In IMS CP Ontology, each element is an instance of a subclass of Element and has a unique rdf:ID attribute. The query in Fig. 3 obtains an rdf:ID and title of all courses available in the curriculum by querying all instances of class Organization of IMS CP Ontology. It should be noted that all the queries are launched through the desktop client and therefore, details of the nRQL are hidden from users. It should be emphasised that users do not need to have any knowledge of nRQL.

```
(retrieve
      (
            ?element
            (datatype-fillers (#!imscp:title ?element))
      )
      (?element #!imscp:Organization)
)
```

Fig. 3. Find all available courses in the curriculum

The result contains the two courses that we chose to implement: "Algorithm" and "Data Structures and Object-Oriented Programming" as shown in Fig. 4. This response will be transformed and shown in the desktop client as 1st level nodes in the Content Navigator. Organization elements link with Item elements via *hasItem* property. Using the rdf:ID attribute of a course, we can retrieve rdf:ID attributes of all Item elements under it using *hasItem* property. Since Item elements also link with each other via *hasItem* property, we can retrieve all Item elements under those elements and so on.

In the IMS CP Ontology, each Item element has an associated Resource element. A Resource element may have any number of File elements. A File element has an associated FileMetadata element. Therefore, an Item element may have any number of FileMetadata elements. In the application, only Organization, Item and FileMetadata elements are shown in the course structure. Organization elements contain course information, FileMetadata elements contain a physical resource's information and Item elements contain all the intermediate contents between Organization and FileMetadata elements.

```
(((?element #!:NTU_CSC105_Organization_2)
        ((:datatype-fillers (#!imscp:title ?element))
         ("Data Structures and Object-Oriented Programming")))
       ((?element #!csc202:NTU_CSC202_Organization_1)
        ((:datatype-fillers (#!imscp:title ?element)) ("Algorithm"))))
```

Fig. 4. Original response from RacerPro to the query in Fig. 3

To retrieve all FileMetadata elements of an Item element, we use three similar queries. The first query retrieves the associated Resource element (using *hasResource* property), the second gets File elements that the Resource element has (using *hasFile* property) and the third fetches the FileMetadata elements associated with those File elements (using *hasMetadata* property).

Fig. 5 shows a part of our curriculum structure. At the root is the curriculum. The first level nodes are the two available courses. The leaf node with title "COURSEWORK ASSIGNMENT" is a FileMetadata element. Other intermediate nodes are Item elements.

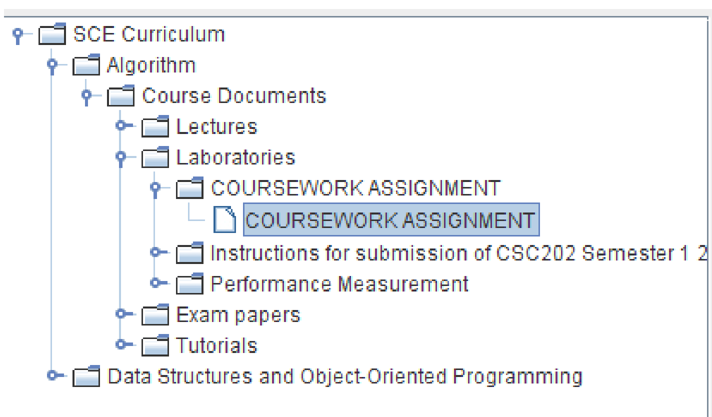

Fig. 5. Content Navigator

When learners double click on a FileMetadata element, a query retrieving datatype properties of the element (the left column of Area B in Fig. 2) will be sent to RacerPro. The response will be transformed and displayed to the learners in the Detailed Information Panel.

4.2 Discovering Related Resources

As we mentioned in the previous section, implicit relationships between resources can be discovered based on links with the Computing Ontology. In i-OntoLearning, we want to focus on the cross-course relationships.

Suppose you are viewing a lecture entitled “Graph Algorithms - Minimum Spanning Trees” and wish to locate resources from other courses relevant to this lecture. As mentioned in section 3.2, we define a property called *relatedTo* as a super property of *isA*, *isPartOf*, *isClassifiedAs* and *uses*. Hence, if this lecture possesses one of these four properties, it will also have property *relatedTo*. We can create a query to find all the instances of FileMetadata that has *relatedTo* properties with the same topic as this lecture and are from a different course. Table 1 shows results of such a query. This includes title, learning resource type, corresponding relationship with Computing Ontology and the topic it relates to. Each row depicts a resource. Topics “Trees” and “Graphs” are instances of topic “Linked Structures” which belongs to the topic “Fundamental Data Structures”. FileMetadata “Minimum Spanning Tree” is defined by the author to have *uses* property with topic “Graphs” and *isA* property with topic “Trees”.

Table 1. List of resources related to lecture “Graph Algorithms – Minimum Spanning Trees”.

No	Title	Learning Resource	Relationship with topic	Topic
1	CPE/CSC 105 Tutorial 8: Trees	exercise	isClassifiedAs	Trees
2	CPE/CSC 105 Tutorial 10: Graphs	exercise	isClassifiedAs	Graphs
3	Java code for List, Stack, Queue and Binary Search Trees	experiment	isA	Trees
4	Trees: Traversal	lecture slide	uses	Trees
5	Binary Search Trees	lecture slide	isA	Trees
6	Other Types of Trees	lecture slide	isA	Trees
7	Trees: Basics	lecture slide	isClassifiedAs	Trees
8	Graphs: Basics	lecture slide	isClassifiedAs	Graphs
9	Graphs: Traversal	lecture slide	uses	Graphs

Relationships between resources may have different meanings due to the two involved properties. For example, the Algorithms course has a lecture on “Hashing” and users may wonder what this technique is for and how it can be applied. The user can pose a query to find resources that use the topic in which lecture “Hashing” is classified (combination of *isClassifiedAs* and *uses*). FileMetadata element “Indexing Techniques” which is a part of a chapter on “Indexing” in the Information Retrieval course would be in the result since it has property *uses* with topic “Hashing”. We can say that lecture “Hashing” is used by resource “Indexing Technique”. Other users may want to review “Linked Lists” that is used by “Hashing” and ask for resources covering this topic (combination of *uses* and *isClassifiedAs*). The result contains lectures “Linked Lists” and “Linked Lists Implementations” from the “Data Structure and Object-Oriented Programming” course.

There are various combinations comprising two sub properties, which correspond to the selected resource and related resources. We divide those combinations into five groups:

(I) **Similar**: This group contains resources that are similar and contain knowledge at the same level of detail.

(II) **General**: This group contains resources that are more general than the selected resources.

(III) **Specific**: This group contains resources that are more specific than the selected resources.

(IV) **Uses**: This group contains resources that the selected resource may use.

(V) **Used by**: This group contains resources that may use the selected resource.

Related resources of the selected resource “Graph Algorithms - Minimum Spanning Trees” in Table 1 can be classified into these groups. As shown in Area C of Fig. 2, group “Similar” has three resources: “Java code for List, Stack, Queue and Binary Search Trees” contains code for Binary Search Tree; “Binary Search Trees” and “Other Types of Trees” are lecture for different types of tree. These three resources belong to the group since they have “*isA* and *isA*” combination with the selected resource. Group “Uses” consists of lecture “Graphs: Basics” and tutorial “CPE/CSC 105 Tutorial 10: Graphs” which covers the “Graph” topic that is used by the selected resource. Lecture “Trees: Basics” and tutorial “CPE/CSC 105 Tutorial 8: Trees” are classified as topic “Trees” and therefore belong to group “General” since the selected resource has property *isA* with this topic. Group “Used By” contains lecture “Trees: Traversal” which has property *uses* with topic “Trees”. The last resource, lecture “Graphs: Traversal”, does not belong to any group because it has property *uses* with topic “Graph” and the combination “*uses* and *uses*” is not considered. In this example, group “Specific” does not contain any resources. Fig. 6 shows the “General” and “Used by” group in the desktop application.

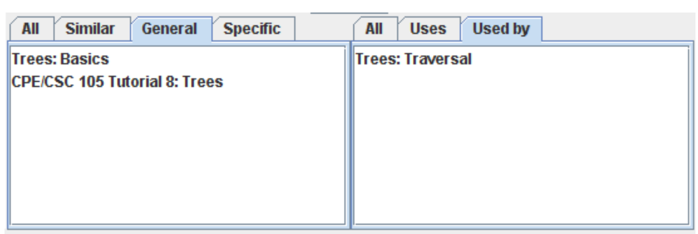

Fig. 6. “General” and “Used by” group

The five groups are divided in two tabbed panes. One pane contains the groups, “Similar”, “General” and “Specific” as well as “All” which combines all three together. The second pane allows learners to focus on “Uses” and “Used by” or both of them under “All”. In i-OntoLearning, conditions can be added on attributes of desired resources. For example, we can add two conditions to the result of related resources of lecture "Graph Algorithms - Minimum Spanning Trees": the filename must contain string "Tree" and the file-size must be at least 100,000 bytes. RacerPro returns the fourth to seventh rows in Table 1. Alternatively, we can retrieve only lecture resource by adding the condition that attribute learningResourceType of Metadata Ontology must contains the string "lecture". This feature can also be exploited to implement an advanced search function. Thus, in addition to datatype and annotation properties, topics can also be used to specify conditions. A query can be defined to find all resources that use "Hashing" and have title that contains the string "Indexing". As a result, RacerPro returns resource "Chapter 4: Indexing" from an "Information Retrieval" course.

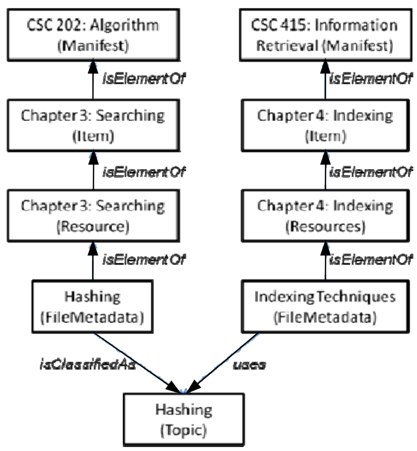

Fig. 7. Relationship between Item elements

Queries have been used to explore the relationships between FileMetadata elements. These relationships can be used to deduce associations between Item elements, since each FileMetadata element is a child element of exactly one Item. As shown in Fig. 7, the relationship between FileMetadata element "Hashing" and "Indexing Techniques" is also applied to Item element "Chapter 3: Searching" of Algorithm course and Item element "Chapter 4: Indexing" of Information Retrieval course. Therefore when an Item element is selected, related items will be shown in the Related Resource Panel of the desktop client.

5 CONCLUSION

The paper describes the ontologies and queries that enable students to explore the relationships between courses in a given curricula. The ontology design maintains a clear separation between metadata and content structure as well as between courses. This ensures that new topics and relationships can to be added without the need to alter the metadata layers. i-OntoLearning is flexible in the number of courses and resources it can support. Navigation through the content structure of resources is facilitated by queries. Moreover, with the use of the Computing Ontology, related resources can be found.

Currently, specific course ontologies are built manually. We intend to build a graphical user interface so that authors can construct these course ontologies without knowledge of an ontology description language. Metadata extractions will be applied to resources to expedite this process, and further metadata properties can be added to the Metadata ontology. Using IMSCP as a framework for content structure modelling facilitates integration with e-learning systems such as BlackBoard [14] that also use IMSCP.

This application based on the Computing Ontology illustrates that learning experience can be enriched. With eight levels of details, the mapping between resources and subjects will be more accurate, which will result in more relevant retrieval of learning resources. Implicit relations between resources can be derived based on properties between topics.

References

1. Henze, N., Dolog, P., & Nejdl, W.: Reasoning and Ontologies for Personalized E-Learning in the Semantic Web. Educational Technology & Society, 7 (4), pp. 82-97 (2004)
2. Saini,P.S., Ronchetti, M.: Deriving ontology-based metadata for e-learning from the ACM Computing Curricula. IADIS International Conference e-Society, Lisbon, Portugal (2003)
3. Ishaya, T.**:** Ontologies for semantic integration of multimedia e-learning services. 3rd ACS/IEEE Int. Conference on Computer Systems and Applications. pp. 883 – 888 (2005)
4. Votis, K., Christos. A., Giotopoulos, K., Likothanassis, S.: An Ontological Service Oriented and Web Searching Mechanism in Distributed e-learning Repositories, Sixth IEEE Int. Conference on Advanced Learning Technologies (ICALT'06), pp 211-215 Kerkrade, Netherlands (2006)
5. Tan M., Goh A.: The Use of Ontologies in Web-based Learning. International Workshop on Ontologies and Semantic Web for E-Learning (SW-EL), Hiroshima, Japan. (2004)
6. Cassel L. N., Davie G., Fone W., Hacquebard A., Impagliazzo J., LeBlanc R., Currie Little J., McGettrick A., Pedrona M.: The computing ontology: application in education. ACM SIGCSE Bulletin. 39 (4) pp. 171-183 (2007).
7. W3C, OWL - Web Ontology Language Overview. http://www.w3.org/TR/owl-features/
8. Instructional Management System. IMS Content Packaging Information Model, Version 1.1.4 Final Specification. October, 2004.
9. Computing Curricula 2005. http://www.computer.org/portal/cms_docs_ieeecs/ieeecs/education/cc2001/CC2005- March06Final.pdf (2005)
10. Dublin Core Metadata Initiative. http://dublincore.org/
11. IMS Meta-data Best Practice Guide for IEEE 1484.12.1-2002 Standard for Learning Object Metadata, Version 1.3 Final Specification http://www.imsglobal.org/metadata/mdv1p3/imsmd_bestv1p3.html
12. Open Archives Initiative/Object Reuse and Exchange http://www.openarchives.org/ore/
13. RacerPro http://agraph.franz.com/racer/racer_features.lhtml
14. Blackboard Learning System. http://www.blackboard.com/