A Semantic Approach to Intelligent and Adaptive Learning Based on Web Learning Objects

Maria Sette and Lixin Tao

Pace University Seidenberg School of CSIS, White Plains, NY 10606, USA
{msette, ltao}@pace.edu

Abstract—Tutoring systems are computer-based learning systems. In order to be “intelligent” they must present “human-like” tutoring capabilities. Tutoring systems should be able to adjust the content and delivery to students’ characteristics and needs by analyzing and anticipating their effective responses and behaviors. There is ongoing research to improve the effectiveness of dialogue and feedback for intelligent tutoring systems. This research aims at the possibility of developing learning materials from an Introduction course in Web Technology as learning objects and how this can be incorporated into a web based adaptive learning and testing system. The approach is to use the current semantic web technologies to model knowledge concepts and the relationships among them, along with a format suitable for computer reasoning and support intelligent learning and personalized learning experience by integrating assessments into learning process.

Keywords; Semantic, Artificial Intelligence, Learning Objects, Knowledge Representation, Adaptive Learning, Intelligent Tutoring Systems

I. INTRODUCTION

Over the past few years there has been an increase in the awareness of a semantically adaptive web-based learning environment. This has been driven by the idea of individualizing tutoring systems for personalized learning, which cannot be achieved at a massive scale using the traditional approaches. Factors that contribute in this direction include the diversity in the target population participating in learning activities, the diversity in the access media and modalities that one can effectively utilize today in order to access, manipulate or collaborate on educational content or learning activities, alongside with a diversity in context of use of technologies. [1]

Shareable Content Object Reference Model [2] provides specifications for implementing e-learning systems and enabling learning object reusability and portability across diverse Learning Management Systems [3]. The LOM in SCORM is used to provide consistent descriptions of SCORM complaint learning objects, such as Content Aggregations, Activities, Shareable Content Objects and Assets so that they can be identified, categorized, retrieved within and across systems in order to facilitate sharing and reuse [4].

II. RELATED RESEARCH

The literature review represents the Learning Object and Knowledge Representation extensions of metadata standards especially concern to educational domain, which are developed by various organizations and researchers.

OntoEdue project [5] emphasized the need of ontology for adaptability and personalization techniques in e-learning domain.

ALOCoM ontology [6]: is designed to generalize the content models and to provide an ontology-based platform to integrate different content models by explicitly defining their structure of LOs. The revised ALOCoM ontology [7] divided into two different parts:

ALOCoM Content Structure ontology is to enabling a formal representation of LOs.
ALOCoM Content Type ontology is to define the educational role of LOs and their components.

CoAKTinG project [8] developed ontology for distributed e-Science through the application of advanced knowledge technologies.

Mohan & Brooks [9] proposed three different types of ontologies related to Learning Objects such as domain ontologies to cover subject area, ontologies that covers learning and teaching strategies and ontologies for structuring of learning objects.

The EUME Onto [10] is an educational ontology that mainly contains concepts related to learning resources, learning design and learning contents.

The SCORM specification provides a limited number of fields for context-related information. But the efforts such as Reusable Learning Objects (RLOs) have not had widespread uptake [11].

The LOFinder [4] is an intelligent LOM shell that provides an alternative approach to enhancing the knowledge representations and enables intelligent discovery of learning objects.
Learning Objects, Learning Object Repositories, and Learning Theory provides best practice for online learning courses [12].

IEEE Standard for Learning Object Metadata specification which provides standards to facilitate search, evaluation, acquisition, and use of learning objects for instance for learners or instructors or automated software processes [13].

III. Problem Definition

The current issues with Learning Object Metadata (LOM), is that it is an XML-based development, which emphasizes on syntax and format and is not semantically driven by knowledge representation. LOM does have the advantage of data transformation and digital libraries, but lacks the semantic metadata to provide reasoning and inference. XML-based markup language is limited in this approach because it causes compatibility problems with existing data applications.

The current e-learning standards where purpose for learning management systems and not for integration with Semantic Web-Based application systems. The most recent standard for IEEE LOM is expressed in Resource Description Framework (RDF) [13].

RDF alone does not provide common schema that helps to describe the resource classes and represent the types of relationships between resources. A specification with more facilities then those found in RDF to express semantics flexibility is needed [13]. A semantic web-base intelligent tutoring system solution can help solve these problems.

IV. Motivation

This research study focuses on the development of ontology for Learning Objects in a semantically adaptive web-base application called PaceJena Tutoring System which will be used to demonstrate how Learning Object Metadata can be extended using Web Ontology Language and Resource Description Framework based development, to emphasizes semantics, knowledge reasoning and inference functions. These functions are necessary for computer-interpretable descriptions, which are critical in the area of dynamic course decomposition, learning object mining, learning object reusability, and autoexec course generation [16]. This research will also identify important relationships for effective knowledge representation and use reasoning for supporting adaptive learning based on web learning objects along with using a prototype to validate feasibility of runtime knowledge representation. We will also investigate and examine the challenges of adaptive personalized learning and learning process for different students and assess learning outcomes.

V. Methodology

This section describes the process in developing an educational ontology for computer science domain in the subject area of “Introduction to Web Technology”.

Before starting to define the ontology, different development methodologies were studied [14]. The methodology that was chosen for this study is METHONTOLOGY [15] that proposes a set of activities to develop ontologies based on its life cycle and the prototype refinement; and 101 Method [16] that proposes an iterative approach to ontology development.

There is not just one correct way or methodology for developing ontologies. Conceptual modeling activity, ontology construction must be supported by software engineering techniques [17]. This research used methods and tools from software engineering to support ontology engineering activities. The ontology development for this research was done using specification and conceptualization.

The specification phase is to acquire informal knowledge about the domain and the conceptualization phase is to organize and structure this knowledge using external representations that are independent of the implementation languages and environments. This section shows how we have adapted different ontology development methodologies to define the specification and conceptualization phases. This also shows how different software engineering techniques were used to define different representations during these phases.

VI. Domain Description

This work was made from scratch using a course from Pace University called “Introduction to Web Technology”. This course ontology was developed using the 101 METHOD and the METHONTOLOGY, which uses intermediate representations for organizing knowledge domain in the conceptualization phase [15].

VII. Scenarios and Competency Questions

The scenarios show problems that arise when people need information that the system does not provide. Besides, the scenario description contains a set of solutions to these problems that includes the semantic aspects to solve them. In order to define scenarios and communicate them to the involved people, templates have been used. These templates were based on those proposed to specify case uses in object oriented methodology [18] (Uschold & Gruninger, 1996). An example is shown in Table 1. The template describes: the name of the scenario, people who participate in the scenario, a brief scenario description, and a list of possible terms related to the scenario. Since this template shows the most important information in a concise way, it is useful when the experts do not have a lot of time to analyze the scenarios.
Scenario: Student Profile.

Actors: Participants of a computer science course called “Introduction to Web Technology”.

Description: The scenario proposed here is a student whom is enrolled in a computer science course. This task gathers information about student’s competency level and according to this information students are asked a series of questions to understand students issues. The intelligent tutoring system will gather this information through students input. Once student’s input has been completed the system will search, query, reason, conduct inference and will output the results. The results may contain the following learning objects; text documents, slide presentations, audios, videos, graphics, diagrams, or link to other resources.

Terms: student profiles, competency levels, intelligent tutoring system, learning objects

Table: 1 Scenario Description

<table>
<thead>
<tr>
<th>Terms</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Competency questions</td>
<td>Proceed from the scenarios. This allows ontology scope to verify if it contains enough information to answer these questions and to specify the student competency level and detail level required for the responses. Besides, it defines expressivity requirements for the ontology because it must be able to give answers using its own terms, axioms and definitions. The scope must define all the knowledge that should be in the ontology as well as those that should not. It means that a concept must not be included if there is not a competency question that uses it. This rule is also used to determine whether an axiom must be included in the ontology or not. Moreover, competency questions allow defining a hierarchy so that an answer to a question may also reply to others with a more general scope by means of composition and decomposition processes. As an example, some of them are shown in Table 2.</td>
</tr>
<tr>
<td>Simple questions</td>
<td>What is your competency level? What is the Introduction of Web Technology topic? What is the Introduction of Web Technology subtopic? What is the Learning Object? What is the Keyword?</td>
</tr>
</tbody>
</table>

Table: 2 Competency Questions

Table: 3 Key Terms

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web Technology</td>
<td>Text Document</td>
</tr>
<tr>
<td>Web Architecture</td>
<td>Tag Names</td>
</tr>
<tr>
<td>HTML</td>
<td>Links</td>
</tr>
<tr>
<td>URL</td>
<td>Elements</td>
</tr>
<tr>
<td>HTTP Protocol</td>
<td>Attributes</td>
</tr>
<tr>
<td>Session Data Management</td>
<td>Cookies</td>
</tr>
<tr>
<td>Web Browser</td>
<td>Hidden Fields</td>
</tr>
<tr>
<td>Computer Program</td>
<td>Query String</td>
</tr>
<tr>
<td>Web Server</td>
<td>Server Side Session</td>
</tr>
<tr>
<td>Application Server</td>
<td>DNS</td>
</tr>
<tr>
<td>Database Server</td>
<td>HTTP Request</td>
</tr>
<tr>
<td>Internet TCP/IP</td>
<td>HTTP Response</td>
</tr>
<tr>
<td>Markup Language</td>
<td>Ports</td>
</tr>
<tr>
<td>Application</td>
<td>Domain Names</td>
</tr>
<tr>
<td>Resources</td>
<td>Web Resources</td>
</tr>
</tbody>
</table>

Fig: 1 Scope of Knowledge Representation
IX. IMPLEMENTING INTRODUCTION TO WEB TECHNOLOGY USING PROTEGÉ 4.2

The tool that was used to implement this domain ontology was Protégé 4.2, because of the fact that it is extensible and provides a plug-and-play environment that makes it a flexible base for rapid prototyping and application development [19]. Protégé ontologies can be exported into different formats including RDF Schema (RDFS) [20], and Web Ontology Language (OWL) [21]. Particularly, we have implemented the Web Technology Ontology in OWL and verified its consistency by using Hermit OWL Reasoner 1.3.6 [22]. It was very useful for determining unsatisfiability problems and their propagation causes. An OWL class is deemed to be unsatisfiable (inconsistent) if, because of its description, it cannot possibly have any instances [23].

To compare the ontology implementation with its conceptualization, graphics using the OWLViz and Ontoviz plug-ins were generated. On the one hand, OWLViz enables the class hierarchies in OWL Ontology to be viewed, allowing comparison of the asserted class hierarchy and the inferred class hierarchy. OWLViz integrates with the Protege-OWL plugin, using the same color scheme so that primitive and defined classes can be distinguished, computed changes to the class hierarchy may be clearly seen, and inconsistent concepts are highlighted in red. Fig. 2 shows the Domain Ontology taxonomy.

On the other hand, OntoViz generates diverse combinations of graphics with all relations defined in the ontology, instances and attributes. OntoViz allows visualizing several disconnected graphs at once. These graphs are suitable for presentation purposes, as they tend to be of good clarity with no overlapping nodes. An example of OntoViz is shown in Fig. 3 which illustrates the main relations and concept instances for Web Technology.

Fig: 2 Domain Ontology Taxonomy

X. VERIFYING AND VALIDATING OF ONTOLOGY

This research proposes to use SparQL to verify and validate the competency questions, which is a SQL like query database. SparQL is a query language and a protocol for accessing RDF designed by W3C RDF Data Access Work Group. As a query language SparQL is a “data-oriented” which only queries the information held in the models. There is no inference in the query language itself. SparQL does not do anything other than take the description of what the application wants, in the form of a query, and returns the information, in the form of a set of bindings or a RDF graph. [24] It treats RDF as data and provides queries with triple patterns and constraints over a single RDF model. Another query language is OWL-QL [25] which was designed for query-answering dialogues among agents using knowledge in OWL. Then, OWL-QL is suitable when it is necessary to carry out an inference in the query.

XI. CONCLUSION

This research is currently in progress and is undergoing iterative updates and changes.

ACKNOWLEDGMENT

This is to dedicate my appreciation and mentor to Dr. Tao, my advisor for his patients, guidance, and inspiration.
REFERENCES


