Towards Best Practices for Semantic Web Student Modelling

Mike WINTER, Christopher BROOKS[1] and Jim GREER[2]
{mfw127, cab938}@mail.usask.ca[1]
greer@cs.usask.ca[2]
ARIES Laboratory, Department of Computer Science,
Saskatoon, Saskatchewan, Canada S7N 5A9

Abstract. Semantic Web applications offer great potential to student modellers who have traditionally struggled with issues of re-use, portability and tight coupling with learning applications. In this paper, we describe our use of ontology languages and e-learning standards to develop a loosely coupled and portable student modelling architecture used in a large-scale, distributed production learning environment. [1]

Introduction

Student modelling systems face a set of challenges when trying to model student activity on real e-learning systems. The collection of student modelling data is time-consuming and requires the development of data structures to represent student activities within the applications of interest. Once student data is collected, it must be converted into a format compatible with knowledge representation and reasoning systems to function as the input for various adaptive systems. Faced with these requirements, student modelling data is often stored in proprietary, hard-to-access formats that don't encourage reuse or distributed study. Additionally, student modelling systems are often tightly coupled with the learning applications they are developed for, rendering them useless when the application is changed or replaced.

Recently, student modelling researchers have begun to adopt technologies, applications and standards from the Semantic Web and e-learning communities to solve the problems mentioned above. Student modellers are developing their domain models and student models using semantic web ontology language such as the Resource Description Framework Schema (RDFS) or Web Ontology Language (OWL) [2][4][13]. Student models developed with a semantic web ontology language have the advantages of formal semantics, easy reuse, easy portability, availability of effective design tools, and automatic serialization into a format compatible with popular logical inference engines. To support loosely coupled student modelling systems, developers are working with e-learning environments that conform to widely accepted e-learning specifications, such as those developed by the IMS Global Learning Consortium[2]. Student modelling systems that are developed using techniques from the Semantic Web and e-learning specifications have the potential for greater relevance and reuse in real learning systems.

---

[1] Funding for this research was provided by the LORNET grant from the National Science and Engineering Research Council of Canada.

[2] http://www.imsglobal.org/

The University of Saskatchewan Advanced Research in Intelligent Educational Systems (ARIES) laboratory has spent the past year using Semantic Web tools and e-learning specifications to develop a loosely coupled and reusable student modelling architecture. This architecture aggregates student data from multiple e-learning applications that have large amounts of use from real students. The Semantic Web middleware application developed to transport the student data from the e-learning applications to interested researchers has been discussed in previous publications [5][2], so in this paper we focus on the details of effective student modelling using web ontology languages and e-learning specifications and provide recommendations for future ontology-based student modelling projects. The layout of the paper is as follows: Section 1 discusses the use of ontology languages for developing domain models, Section 2 discusses the process of collecting and representing student model data with the use of standards-based e-learning tools and ontology languages while Section 3 gives an overview of the deployment of our student modelling system in a production environment. Finally, Section 4 provides conclusions and discussion on future work.

1. Towards a Best Practice for Ontology-based Student Modelling

1.1 Introduction to Semantic Web Student Modelling

Ontology languages are used to structure and share knowledge, especially for the use of software applications capable of reasoning that require explicit definitions of concepts and the relationships between those concepts. Evolving from various frame-based representation languages, web ontology languages are being developed as part of the World Wide Web Consortium (W3C) Semantic Web project. The W3C's recommended specification for ontology languages is the Web Ontology Language (OWL), which has three different varieties: OWL Lite, OWL DL and OWL Full. Lite to DL to Full, provide different levels of logical expressiveness, with Lite being the least expressive and Full being the most expressive. The logical semantics of OWL DL (and Lite, which is a subset of DL) are based on a description logic, which is a decidable subset of full first-order logic. This means that all inferences available in an OWL DL ontology can be computed. That is not the case for OWL Full, which is not decidable, and has little to no application reasoning support available. For those reasons, most users of OWL strive to keep their ontologies in OWL DL to ensure maximum utility, ease of development and reuse.

An increasing number of student modelling systems using these ontology languages to specify the structure and properties of their associated student models. Typical approaches are found in [4] where OWL ontologies for a human-computer interaction course are automatically generated from a dictionary and then annotated by hand to fully reflect the course content, and in [11] where IMS Learning Design functions are annotated with OWL ontologies representing an individual's domain knowledge. In this section we discuss our experience of developing a set of student model ontologies that maximize the benefits promised by web ontology languages: extensibility, portability, and inferential power.

1.2 Effective Ontology-based Student Modelling

It is not immediately obvious how to construct an effective production student model using existing web ontology languages. We eventually decided to use OWL DL as our ontology language of choice because of its functionality, tool support (in particular, the Protégé[3] development tool) and status as an official W3C recommendation. In terms of the general structure of our student model ontology, our advice is to separate the ontologies into three broad areas: those that that represent student characteristics, those that encapsulate abstract domain knowledge and relationships, and those that model the concrete subset of the domain taught in particular course along with the learning resources available in those courses. This is similar to the approach taken by other researchers who have used ontology languages to develop student modelling systems [13][8]. By loosely coupling the three different types of ontologies, a student modelling application is better able to react to changes in course subject matter, learning material and student type, which often happens on a semester-to-semester basis in practice. Decoupling the abstract domain ontology of an area of study from the ontologies representing the particular topics and learning resources associated with a course is a particular useful practice. The separation allows a generally static domain ontology to be developed that can be reused across multiple courses teaching different aspects or levels of difficulties of the same area of study even as the particular resources and topics in a given class change rapidly.

Separating the general taxonomy of the domain from the particular instances of the topics being taught in a course also provides a solution to a problem facing ontology developers using the OWL DL and OWL Lite variants: representing classes as property values [6]. When developing an ontology using OWL, one cannot have classes as property values (with the exception of the *rdf:Type* property) without moving the ontology into the OWL Full variant, which is not desirable for the reasons stated above. However, a common statement student modellers want to make is of the general form "*user* knows *topic". If *topic* is represented in the ontology as a class, then the ontology will be in OWL Full. Separating out the course-specific instances of topics from the classes in the taxonomy that represent the topics in the abstract allows for the ontology to stay in OWL DL without the awkward, maintenance-heavy artifice of some of the Semantic Web Best Practices and Deployment Working Group's solutions to the classes-as-property-values problem [6]. Using such a separation also makes intuitive educational sense for a reusable domain model: if a topic is being taught in a first-year and a third-year course, statements in the ontology saying that students from the respective courses can know the topic at an equal level are not likely accurate (although you could also develop an expressive set of other properties to capture the depth of knowledge learned, as discussed in Section 2).

1.3 Capturing Useful Pedagogical Relationships in the Domain and Course-Specific Ontologies

The most straightforward way in OWL DL to separate the classes that represent the domain model from the instances that represent the topics being taught in a particular course is to use the *subClassOf* property to model the relationships between classes in the abstract domain model and the *instanceOf* property to connect the concrete course topics to the classes in the abstract domain model. Having a domain ontology constructed using these properties provides only generalization/specialization relationships in the general taxonomy and type information for the topic instances of the course. Figure 1 shows a section of our abstract domain model for the HTML domain, which is constructed only with subclass (*is-a)* relationships. Abstract domain models should fully represent all of the topics in a domain so they can be reused between the different courses that teach the domain they represent.

---

[3] `http://protege.stanford.edu/`

However, a richer pedagogical vocabulary is needed to accurately represent the educational relationships between the concrete topics in the individual course ontologies.

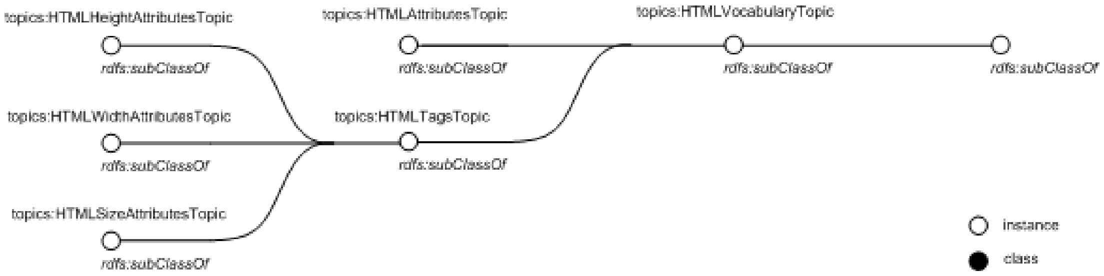

Figure 1: Fragment of an Abstract Domain Ontology

Our first attempt at using a more expressive educational ontology was to develop an ontology representing the granularity hierarchy formalism, which provides properties representing aggregation and specialization relationships between topics [7]. In the granularity hierarchy ontology, a K-Cluster represents a particular aggregation of topics while an L-Cluster represents a particular specialization of a topic. A topic can have more than one K-Cluster and/or L-Cluster relationship. While the aggregation relationship proved to be a valuable addition to our domain models, we found that granularity hierarchies still did not provide the necessary precision to model all of the different possible relationships between topics in a course, including strong and weak prerequisites.

Another reason to move beyond the granularity hierarchy ontology to describe our course-specific topic ontologies was our larger goal of using standard and widely-accessible tools whenever possible to maximize the portability and extensibility of our student modelling system. There are several widely-used metadata standards that we considered using. An approach taken by Muñoz and de Oliveira in their development of ontologies for the AdaptWeb Knowledge Space project is to model both the domain model and the course topics (which they refer to as a *Content Knowledge Ontology)* with an application profile (instantiated subset) of an RDF binding of the IEEE Learning Object Metadata (LOM) specification [1][3][13]. The LOM specification is a standard developed to describe the metadata associated with a given learning object. It has a rich set of elements to describe learning objects and their use, including *isPartOf* and *hasPrerequisite* properties. However, we decided against using LOM, mainly because it is intended for describing the connections between material learning objects, not the intrinsic pedagogical relationships between the topics presented in a course. Also, the RDF binding of IEEE LOM used by Muñoz and de Oliveira is in OWL Full (Muñoz and de Oliveira used the DAML+OIL ontology language for this particular project, rendering that particular concern irrelevant for them).

The ontologies we decided to use as the basis of our course topic ontologies are from the W3C's Knowledge Organization Systems and the Semantic Web (SKOS) project: SKOS Core [14] and SKOS Extensions [15]. The SKOS family of ontologies was specifically developed to describe taxonomies and classification schemes and thus has an excellent variety of properties to describe the relationship between topics in a course. We developed OWL DL compliant versions of both the Core and Extensions ontologies and used them to develop the topic ontologies of particular course offerings[4]. The Core and Extensions ontologies provide several different variations of aggregation and specialization relationships as well as a class called a *ConceptScheme* that organizes related topics. Our use of the SKOS ontologies in modelling the content of a first-year course teaching HTML is illustrated in Figure 2: we have

---

[4] `http://ai.usask.ca/mums/schemas/2005/01/27/skos-core-dl.owl`
`http://ai.usask.ca/mums/schemas/2005/01/27/skos-extensions-dl.owl`

a *ConceptScheme*, *HTMLConceptScheme*, that represents all of the topics being covered in the course, and all the topics covered in the course are related to the *HTMLConceptScheme* instance by the *inScheme* property (not illustrated in the figure for space reasons). We then model the relationships between topics in the course ontology by using the aggregation and specialization properties provided by SKOS: *cmpt100:HTMLAttributesTopic* is *narrower* than *cmpt100:HTMLVocabularyTopic*, which indicates a specialization relationship, while *cmpt100:HTMLVocabularyTopic* is *relatedHasPart* with *cmpt100:HTMLHyperlinksTopic* which indicates an aggregation relationship between the two topics. All of the topics in the course ontology (represented here by the *cmpt100* namespace) are linked to their respective classes in the abstract domain map by *instanceOf* relationships.

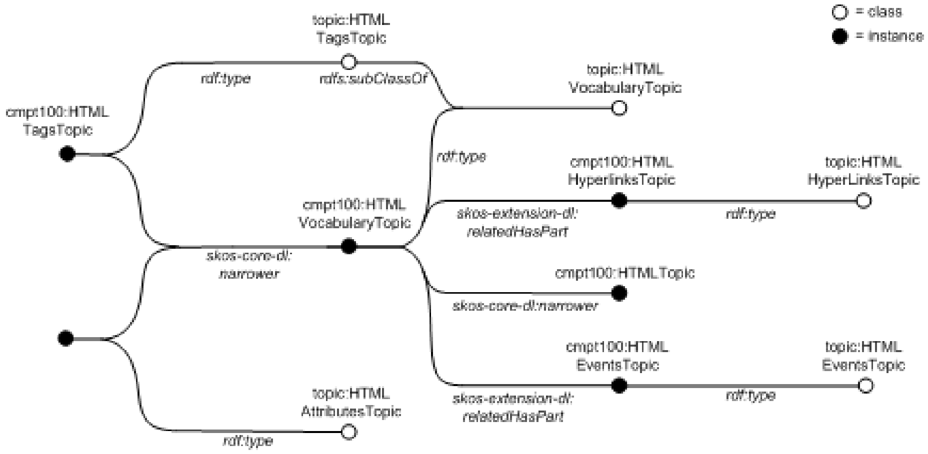

Figure 2: Fragment of an Abstract Domain Ontology with an Attached Concrete Course Ontology that uses SKOS Properties

In this section we demonstrated how we decoupled abstract domain ontologies from course ontologies that only model the topics being taught in particular courses while using the OWL ontology language and staying in its fully logically decidable subset, OWL DL. By using our own DL-compliant binding of the SKOS ontologies we are able to model rich pedagogical relationships between the topics in any given course ontology while still preserving a loosely coupled relationship between the ontologies of different courses that teach the same domain by way of their relationships with OWL classes in the abstract domain map.

2. Modelling Student Competencies and Behaviour

Once the abstract domain and concrete course ontologies are developed, the next step in completing a full student model is to add ontologies about student behaviour and competencies and to develop an effective and portable method to capture student information to populate those ontologies with data. Working towards our goals of maximum reuse and portability, we first examined a number of different standardisation and specification activities taking place in the area of modelling learner competencies. Notable amongst these are the ISO and the IEEE through their work on Public and Private Information (PAPI) for Learners[5], and the IMS Global Learning Consortium and their work on the Learner Information Package (LIP) [11]. These specifications tend to provide containers for learner information as opposed to definitions of what learner information is.

---

[5] `http://jtc1sc36.org/`

For instance, both of these schemes allow for the collection of student marks, but nether provides a schema by which to represent student marks. In this way they leave the definition of useful pedagogical content to other specifications, many of which are ill-defined or very general in scope.

Our goal was to develop an ontology that contained an extensive set of educational relationships that could be expressed as ontological properties connecting students with topics in our course-specific topic ontologies discussed in the last section. To this end, we developed an OWL DL ontology[6] that contains the education relationships outlined by Anderson et al. [9]. This variation on Bloom's taxonomy is a two dimensional model that captures both the kind of knowledge gained in a learning experience (e.g. conceptual knowledge, procedural knowledge, etc.) as well as the cognitive processes the student demonstrated in that learning experience (e.g. remembering, understanding, applying, etc.) . We linked in this Anderson-style ontology with our course topic ontologies by making the range of competency statements appear as topics in the topic-course ontologies.

To populate our student competency ontologies with data about real students, we wanted to use standards-compliant e-learning tools so that both our test questions and student competency ontologies could be easily portable. To this end, we developed our test questions to conform to the IMS QTILite specification [12]. This specification describes a data model and XML-based binding for representing questions and tests in a vendor-neutral manner. The model provides ample semantics for representing content, evaluation, and feedback to the learner, but provides no way of associating outcomes of a test with competencies. To connect the test answers to our student competency ontology, we develop a test-specific ‘glue’ ontology that does the work of connecting QTILite answers to statements about student competencies from the Anderson ontology. Figure 3 shows an example segment of a student model that contains a competency statement derived from a QTILite-compliant testing tool.

By adding outcome semantics to individual question/answer pairs, we are able to create fine grained models about a learner's knowledge state. Further, instead of one "correct" answer and many "wrong" answers, we are able to associate any pieces of demonstrated learning with any question/answer pair. While our current tests only associates knowledge statements with one best answer for each question, our loosely-coupled format also allows us to test different levels of knowledge (represented as a collection of answers) within one question. Further, a quick analysis of all of the possible answers for a question, and their associated educational outcomes, allows us to make statements about what knowledge a student has failed to demonstrate in the test, or about the likely misconceptions the student has, given the answer (the classical ‘bug library’).

The final components of our learner model are ontologies that represent the students and the applications they use. Our student ontology is currently very simple, with just the capacity to uniquely identify a student, as we prefer to keep information about students loosely coupled. In the future, however, the ontology may be expanded to include information about a student’s learning style, demographic information or any other factors that are intrinsic to the student. Our application ontologies are more complex, as they model all of the interesting interactions a student can have with our e-learning applications. For example, our message board ontology contains properties to describe a student’s posting of a message with the composition time, the reading of a message with the dwell time, the changing of a category, and many more. These events are not currently translated into any Anderson-style statements about student competency, but they are currently being used for visualization and data mining projects.

---

[6] `http://ai.usask.ca/mums/schemas/2005/01/27/anderson.owl`

3. Implementation and Deployment

In this paper, we have emphasized a loosely coupled architecture for ontology-language based student modelling that relies heavily on accepted standards and available tools. This approach was refined over a year-long period of developing ontology-based learner models for students enrolled in a first-year Computer Science course that is offered online at the University of Saskatchewan. Initially, we developed RDFS ontologies that represented every topic in every module of the online course, ranging from the History of Computing to Advanced HTML and Javascript programming. Our initial ontologies contained over 1000 different topics and 1200 granularity hierarchy relationships between the topics [2], as well as around twenty-five QTILite-compliant quizzes embedded into the online courses with over one-hundred questions whose answers were mapped to our topic ontologies.

We immediately ran into problems in the first offering of the course, as the content and organization of the course changed over the semester leaving us unable to update our topic ontologies and questions rapidly enough to permit deployment on the course delivery system. This immediately exposed two problems in our ontology development system. First, our ontology development and maintenance “environment” (Wordpad and gvim) provided no support for rapidly building ontologies or checking their semantic and syntactic correctness. Second, changes in the topic ontology of the course left us with the problem of how to properly maintain the domain knowledge we had invested in modelling, while also storing the knowledge about the differences in the domain material and student behaviour associated with the different offerings of the course. Developing a solution for the second problem led us to the conclusion, helped by the discussion in [13] and [8], that the general domain ontology and the course-specific ontologies should be decoupled, as discussed in Section 1. To solve the first problem, we began to use the Protégé ontology development tool, which is a very mature development platform as well as the core of a large user, plugin and development community. Due in main part to the W3C’s recommendation of OWL, a sizable part of the Protégé community is focused on the development of OWL ontologies using the OWL Plugin. A crucial factor in Protégé’s popularity is its ability to communicate with logical inference engines, such as Racer, within the development environment. This feature allows developers to check the semantic and inferential correctness of their ontologies as they develop them, and also provides a powerful incentive to stay within the OWL DL language. The ability to use Protégé with the OWL plugin to develop and maintain our ontologies and W3C’s recommendation was enough to convince us to convert our ontologies to be in OWL DL.

Currently, we have reduced our initially ambitious goals of trying to focus on maintaining domain, course topic models, and QTILite compliant questions for an entire course, to focusing on two (of twelve) modules within the online course (Introduction to HTML and Programming Languages). This will reduce our overhead as we refine our ontology development process. In addition to the highly structured ontologies and competency data reported in this work, our student modelling repository also contains tens of thousands of ontological statements about student behaviour for hundreds of anonymized undergraduate Computer Science students who use our production e-learning systems, which include the iHelp message board and chat system as well as the online course delivered with the iHelp LCMS [2][5].

4. Conclusions and Future Work

In this paper, we presented recommendations on how to construct an ontology-based learner model, backed by our experience of trying to model students within a real, constantly evolving distributed e-learning environment. We described how we decoupled the abstract domain ontology from the concrete topic ontology representing how the domain is taught in individual offerings of courses. While this approach is somewhat similar to that found in [13] and [8], we instead used the OWL ontology language and the SKOS classification ontology, both endorsed by the W3C, to increase the portability, ease of development, and reuse potential of our learner models. Further, we formalized an educational taxonomy proposed by Anderson et al. to map answers on QTILite-compliant tests in a production online course to statements about student knowledge of topics in our course topic maps, as well as gathering large amounts of information about students' behaviour on various e-learning applications.

In the future, we aim to further refine our ontology-based student models in response to our own experiences and those of the larger student modelling community. With our focus based on ontology-based modelling and the RDF data format, we did not spend large amounts of time analyzing standards strongly associated with XML, such as XML Topic Maps, IMS/IEEE RDCEO and the IMS-LD standard (some discussion on this topic can be found in [10]). Learning to apply these standards in future development would likely be beneficial.

References

[1] Nilsson, M., Palmer, M., Brase, J. *The LOM RDF Binding – Principles and Implementations.* The 3rd Annual Ariadne Conference, 20-21 November 2003.

[2] Winter, Mike, Brooks, Christopher, McCalla, Gord, Greer, Jim and O'Donovan, Peter. *Using Semantic Web Methods for Distributed Learner Modelling*. Proceedings on the Workshop on Using the Semantic Web in E-Learning at the 3rd International Semantic Web Conference, pp. 33-26, 2004.

[3] Nilsson, M. *IEEE Learning Object Metadata RDF Binding* [web page]. May 2001. http://kmr.nada.kth.se/el/ims/metadata.html

[4] Kay, Judy and Lum, Andrew. *Ontologies for Scrutable Learner Modelling in Adaptive E-Learning*. Proceedings of the SWEL Workshop at Adaptive Hypermedia 2004. 2004, pp. 292-301

[5] Brooks, Christopher, Winter, Mike, Greer, Jim and McCalla, Gord. *The Massive User Modelling System (MUMS)*. Proceedings of Intelligent Tutoring Systems 2004, pp.635-645.

[6] Noy, Natasha (editor) Stanford University, Stanford, US. – working draft [web page] *Representing Classes as Values on the Semantic Web*. July 21, 2004. http://www.w3.org/TR/swbp-classes-as-values/

[7] McCalla, G. I., Greer, J. E., Barrie, B., and Pospisil, P. *Granularity Hierarchies*, International Journal of Computers and Mathematics with Applications (Special Issue on Semantic Networks), Vol. 23, pp. 363-375, 1992.

[8] Bouzeghoub, Amel, Defude, Bruno, Ammour, Salah, Duitama, John-Freddy and Lecocq Claire. *A RDF Description Model for Manipulating Learning Objects*. Proceedings of the 4th IEEE International Conference on Advanced Learning Technologies. 2004, pp. 81-85,

[9] Anderson, Lorin W. and Krathwol, David R. (eds.). *A Taxonomy for Learning, Teaching and Assessing. A Revision of Bloom's Taxonomy of Education Objectives.* 2001. Addison Wesley Longman Inc.

[10] Dolog, Peter and Nejdl, Wolfgang. *Challenges and Benefits of the Semantic Web for User Modelling.* Proceedings of the Adaptive Hypermedia Workshop at WWW 2003.

[11] Smythe, Colin, Tansey, Frank, Robson, Robby. IMS Learner Information Package Information Model Specification, Version 1.0.

[12] Smythe, Colin, Shepherd, Erik, Brewer, Lane, and Lay, Steve. IMS Question & Test Interoperability QTILite Specification, Version 1.2. February 11, 2002.

[13] Muñoz, Lydia Silva, de Oliveira, Josגé Palazzo Moreira. *Applying Semantic Web Technologies to Achieve Personalization and Reuse of Content in Educational Adaptive Hypermedia Systems*. Proceedings of the SWEL Workshop at Adaptive Hypermedia 2004. 2004, pp. 348-353

[14] Miles, Alistair, Rogers, Nikki and Becket, Dave (editors) W3C [web page] *SKOS Core Ontology.* Feb. 02, 2004. http://www.w3.org/2004/02/skos/core/

[15] Miles, Alistair (editor) W3C [web page] Feb. 02, 2004. http://www.w3.org/2004/02/skos/extensions /