

Adaptive Web-Based Courseware Development using Metadata Standards and Ontologies

Lydia Silva Muñoz¹, José Palazzo Moreira de Oliveira¹

¹ Federal University of Rio Grande do Sul - Instituto de Informática (II/UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil
{lydia, palazzo}@inf.ufrgs.br
<http://www.inf.ufrgs.br>

Abstract. Nowadays, the web enormously facilitates the information distribution and communication among teachers and students setting the stage for the development of Web-based courseware. In this context, adaptability is a key feature in order to obtain educational environments capable of giving the adequate instruction for the specific learning profile of each individual learner. Using the Web as the ubiquitous repository of educative content, standards for metadata to describe resources on the e-learning domain must be used in order to enable interoperability and reuse of learning content. To enable intelligent behavior in building complex learning objects “on demand” customized for an intended audience, ontologies can be used to represent the knowledge the system has on the domain to be taught and on the student profile. This paper briefly describes the main steps taken in developing an Adaptive Web Training Environment, delving into the definition of a new *application profile* of the Learning Object Metadata (LOM) standard and the Domain and Student Knowledge Models design and implementation as Web ontologies.

1 Introduction

To achieve adaptability in an educational hypermedia system, issues on modeling the student’s profile, and the knowledge about the domain to be taught must be taken into account [1][2][3][4][5]. The student’s profile provides the system with the knowledge about the student’s learning goals, domain knowledge, background, cognitive learning style and hyperspace experience, while the knowledge about the domain to be taught lets the adaptive system “know” what is presented on a particular page or page fragment of the instructional material by means of metadata annotations describing it. Then, dynamic content sequence generation tailored to a student profile can be made using the knowledge the system has about the student profile and the system’s learning domain. Metadata can be viewed as concise information containing the most significant attributes of an object. In our approach, the fundamental metadata describing the instructional material is partial generated automatically and stored in a web ontol-

ogy. Ontologies are the description of the entities, relations and restrictions of a domain, expressed in a formal language to enable machine understanding. Tom Gruber [8] has defined “an ontology is a formal, explicit specification of a shared conceptualization”. Additionally, ontologies to be used on the Web need to be supported by established web languages to facilitate interoperability and take advantage of existing tools. In order to make metadata understandable to the Web community, classification schemas defining vocabularies in an unambiguous way must be available on the Web forming *namespaces* [20], which give well understood semantics for the terms used in metadata descriptions. Interoperability and reuse of the learning content can be achieved by having metadata describing it in terms of an established standard metadata model envisaged to describe e-learning content. An *application profile* [10][11] of a standard metadata model is a version of the standard tailored to the requirements of a particular application. This paper describes the main steps taken in creating the Adaptive Web Training Environment AdaptWeb [7], delving into the definition of a new application profile of the Learning Object Metadata (LOM) standard and the Domain and Student Knowledge Model design and implementation as Web ontologies. These ontologies are available at: <http://www.inf.ufrgs.br/~tapejara/Ontology/Generated/AWOntology.daml> and <http://www.inf.ufrgs.br/~tapejara/Ontology/Student.daml>. The AdaptWeb is an in-progress research project supported by the CNPq, Brazilian Research Council, and presently in operation at the Federal University of Rio Grande do Sul. The remainder of this work is structured as follows: section 2 gives an overview of adaptive hypermedia systems, section 3 resumes the main steps taken in the development of the proposed environment, section 4 shows the constructed Knowledge Space model, section 5 explains how the metadata concerning the instructional content is generated, section 6 shows the constructed student model, section 7 briefly explains the adaptive content selection, section 8 shows the proposed system architecture and section 9 is a short conclusion.

2 Adaptive Hypermedia Systems

According to [4] an adaptive hypermedia system has two interconnected networks or “spaces”: a network of elementary knowledge elements forming the *Knowledge Space* and a network of hypertext pages with educational material forming the *Hyperspace*. Links relating the Knowledge Space and the Hyperspace let the adaptive system “know” what is presented on a particular page of the educative content. As stated by authors in [5] the instructional contents’ adaptation to the student’s individual characteristics implies that the system must: (i) recognize the cognitive patterns of each student’s learning and its pedagogical implications; (ii) know the training knowledge space being proposed, (iii) know the instructional material organization, and (iv) be able to dynamically generate the best suited selection and sequence for each student in a particular course stage.

In the AdaptWeb research project, the first such requisite is achieved by using a student model giving the system knowledge about the student's background, learning goals, domain knowledge, hyperspace experience and style of learning. For the system to know the training knowledge space being proposed and the instructional material organization, a machine understandable knowledge model of the domain to be taught was constructed, and the connections between this model and the instructional content were implemented. At the run-time, a teaching strategy to guide dynamic content sequence generation can make it possible to select the specific contents to be presented to the student using the knowledge the system has about the student profile and the system's learning domain.

3 General View

The main steps taken in the development of the AdaptWeb project are shown in figure 1. On the one hand, research was done concerning the different Cognitive Styles of Learning (CSL) students may have, which were described in [5][6]. Based on those CSL, a GUI interface was developed to enable the student himself to categorize his CSL, set his learning goals and characteristics of his work environment and the kind of course he wants to take. At run-time, the student model is updated taking into account the learning activities of the student. More details of the student model are given in section 6. On the other hand, to make an interoperable system, an application profile [10] [11] of the Learning Object Metadata (LOM) standard [9] suitable to the needs of the application educative context was developed, as explained in section 4. A Knowledge Space model based on such a standard application profile was built containing the knowledge the system has about the domain to be taught, as explained in section 4. This model was encoded as a web ontology. An authoring tool was implemented to facilitate the content creation, guiding the authoring process to relate educative content by semantic properties that define the hierarchical structure of the material to a given discipline, and the possible adaptation to pre-defined learning goals and general knowledge background of students. A model that follows a hierarchical approach resembling the structure of a textbook with chapters and sections is defined to structure the educative content. At authoring time the parts that compose a discipline are created and related. Then, learning objects representing topic explanations are related to exercises, examples and complementary material supporting the learning. Also, indications about the degree of difficulty and other characteristics of each learning object are declared. Adaptation to learning goals and knowledge background is made by defining at authoring time what the most adequate parts of the discipline material to be given to students with specific pre-defined learning goals and knowledge background are. For example, Computer Science, Engineering and Mathematics courses customizing the Numerical Methods discipline were defined to be directed to students who have a background knowledge determined by their graduation in such fields. The created instructional content is stored into the system Hyperspace taking advantage of the XML language facilities to represent hierarchical structures. The structure of the

educational content is based on the existence of an XML file for each topic in the Knowledge Space Model. Each XML file representing a topic entails the HTML pages whose content is concerned with this topic. A special XML file is available for each discipline, relating the topic's hierarchical structure to the discipline, and the prerequisite relation among them. Since we have a structured Hyperspace, it is possible automatically to connect the knowledge space with the hyperspace instances by making the semantics behind XML structures explicit, and inferring some metadata elements from others. A wrapper aware of the structure and the context of the hyperspace instances creation automatically generates the metadata describing what the Hyperspace instances contain in terms of the Knowledge Space, and how to use them. This wrapper is executed each time educative content is created. In order to recommend the use of any learning resource discovered on the web and considered suitable to support the explanation of a given topic, it is possible to augment the metadata by the ontology edition. Issues on metadata generation are better explained in section 5.

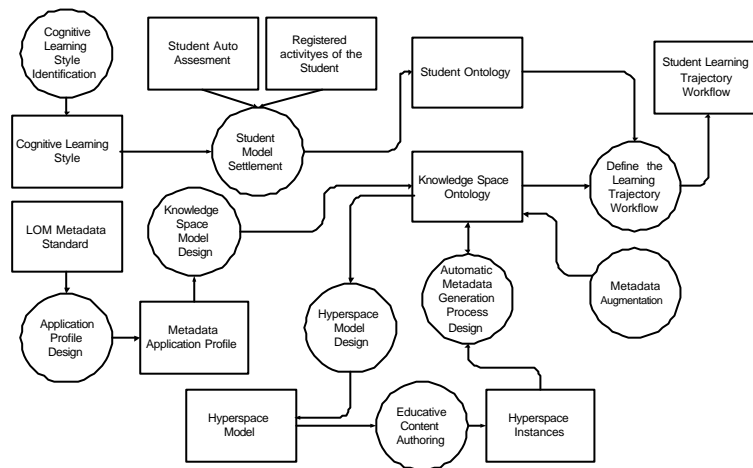


Fig. 1. Main steps in the AdaptWeb project development

Finally, a learning trajectory tailored to the student's profile as a workflow instance can be dynamically generated based on the metadata describing the educative content and the current student model. Also, navigation techniques as direct guidance and hiding links are used to adapt the content presentation to the student's hyperspace experience.

4 Knowledge Space Model

The AdaptWeb Knowledge Space Model describes the structure and semantics of the system learning content. Metadata based in the LOM standard is used. A formal web ontology provides the support so that the educative content composition to achieve

adaptability is possible. Next subsections treat the use of standards in educational metadata on the web, which is the foundation of having a Knowledge Space Model that enables interoperability. Finally, the structure of the constructed model and the use of the LOM properties refinement mechanism to construct the AdaptWeb application profile are explained.

4.1 Metadata in the e-learning domain

Metadata can be viewed as information about the most relevant attributes of an object. It helps in the tasks of discovering, managing and using objects without the need to read, view or explore them directly. To make metadata understandable to the Web community, metadata schemas defining vocabularies in an unambiguous way must be available on the Web forming namespaces which give standard, well understood semantics for the terms used in metadata descriptions. One of the most common metadata schemas used to describe digital resources is the Dublin Core schema – DC [12], which defines the semantics of 15 properties intended to be used in descriptions of cross-disciplinary resources. For the educative domain the Learning Objects Metadata Standard – LOM [9] developed by the IEEE is used. LOM defines the semantics of more than 80 properties to describe metadata in a hierarchy given by the following categories: 1-General, 2-Lifecycle, 3-Meta-Metadata, 4-Technical, 5-Educational, 6-Rights, 7-Relation, 8-Annotation and 9-Classification. Such a standard facilitates the search, evaluation, acquisition and use of learning objects by learners, instructors or automated agents. An application profile [10][11] is the combination of existing schemas into a package that is tailored to the functional requirements of a particular application while retaining interoperability with the original base schemas. Part of this adaptation may include the elaboration of local metadata elements that are important to given community or organization, but which are not expected to be important in a wider context. Several application profiles as ARIADNE [27], Heal-Health Education Assets Library [28], CanCore – Canadian Core Learning Resource Metadata [29], MERLOT [30] and SingCore [31] were developed in order to give more specific semantics to a subset of the LOM specification appropriate for the needs of particular learning contexts, and some crosswalks exist in order to provide mappings between them.

4.2 Application profile implementation

There are several possibilities to represent LOM metadata descriptions on the Web. One is to use an XML binding, i.e. the XML language [13] with a specific XML Schema [15] designed to validate documents with LOM descriptions of a certain Learning Object. Another alternative is the use of an RDF binding, i.e. use RDF [14][32] as the language to represent statements with assertions about the value of LOM properties to a given resource. RDF statements are triples with a subject indicating the resource being described, a property and a value for this property for the described resource (the property value is eventually another resource). We selected RDF to represent LOM descriptions since RDF offers a standard way to combine different

vocabulary definitions supported by RDF Schemas [16], while XML documents have only the semantics resulting from their pre-defined tree structures. For example, we use the Creator property of the Dublin Core schema [12] to identify an Entity (eventually a person) who contributes to the creation of a learning object, and the property FN of the Vcard schema definition [18] to indicate the formal name of such an entity. We use the RDF binding described in [17] as the syntactical representation of the LOM standard. Table 1 shows the subset of LOM properties used as attributes describing each learning object, while the properties describing relations among the learning objects are treated in subsection 4.4. The corresponding abbreviations for namespaces used are vcard: <http://www.w3.org/2001/vcard-rdf/3.0/>, dc: <http://purl.org/dc/elements/1.1/>, dcterms: <http://purl.org/dc/terms/>, and lom-edu, lom-tech, lom-meta stand respectively for documents lom-educational, lom-technical and lom-metametadata in <http://kmr.nada.kth.se/el/ims/schemas/>.

Table 1. LOM properties used in the AdaptWeb application profile.

LOM Property	RDF binding	Description
1.3- Language	dc:language using dcterms: RFC1766	Language used to communicate to the intended user.
1.4- Description	dc:description	Description of the content of the Learning Object.
1.5- Keyword	dc:subject with a literal value	A keyword describing a topic in the Learning Object.
2.3- Contribute	dc:creator and lom:Entity using Vcard:FN	Entity (people, organization) that contributed to the state of the Learning Object.
3.3- Metadata Schema	lom-meta: metadataScheme	The specification used to create the metadata instance.
4.3- Location	lom-tech:location	Location of the Learning Object.
4.4- Requirement	lom-tech:requirement	Technical capabilities necessary to use the Learning Object.
5.1- Interactivity Type	lom-edu: interactivityType	Predominant mode of learning of the Learning Object. Ex: expositive, interactive
5.2- Learning Resource Type	lom-edu: learningResourceType	Specific kind of Learning Object. Ex: Exercise, Example.
5.8- Difficulty	lom-edu:difficulty	How hard it is to work with the Learning Object. Ex: Easy, Difficult.

4.3 Knowledge Space Structure

Each piece of instructional content in the hyperspace is described in the Knowledge Space structure by metadata providing the information for find and reuse it. The Domain Knowledge Ontology implements the Knowledge space structure. It contains the

knowledge describing the pieces of content in the hyperspace and the rules to correctly assemble them in order to compute complex learning objects adequate to the profile of each student. Follows is the description of its classes and relations. Figure 2 shows a graphical representation of the Domain Knowledge Ontology. The abbreviations for namespaces used are *awo*: <http://www.inf.ufrgs.br/~tapejara/Ontology/Generated/AWOntology.daml> and *daml*: <http://www.daml.org/2001/03/daml+oil#>.

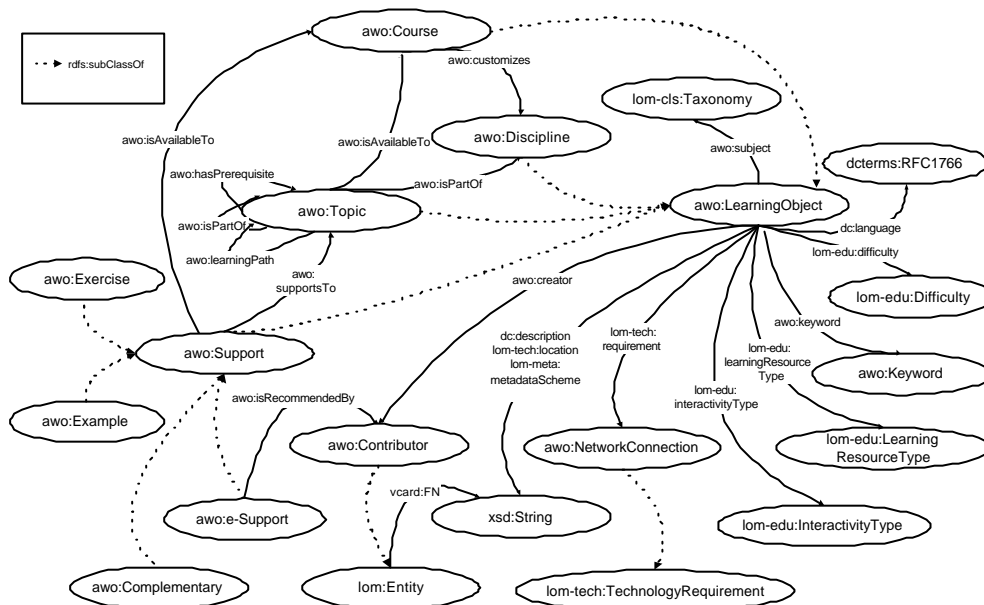


Fig. 2. Domain Knowledge ontology

Instances of class *awo:Topic* represent the explanation of some concept or idea, *topics* are considered to be part of the disciplines learning content. Generally, the explanation of a topic comes associated with some examples, exercises and complementary material supporting the topic study, which are represented in the classes *awo:Example*, *awo:Exercise*, *awo:Complementary* and *awo:e-Support*. Additionally, a topic may have sub-topics giving more specific and detailed explanations related by the *awo:isPartOf* relation, i.e. they are considered part of the explanation of the prime topic. The order in which the topics must be presented according to learning purposes is given by the *awo:learningPath* relation. The class *awo:Course* contains customizations of a discipline to be directed to students with common background knowledge and learning goals. In the case study used, we defined Computer Science, Engineering and Mathematics courses customizing the Numerical Methods discipline for students graduated on these fields. The learning content of a course is composed of the discipline topics that are considered adequate for the intended audience of the course, i.e. the topics *available* for the course. The Exercises, Examples and Complementary mate-

rial supporting a Topic explanation may or may not be available for a course even if the topic they support is available. Generally, the educative material is created in the system context, and an element of the class *awo:Contributor* is indicated as the author of the created learning object. On the other hand, a paper or some other kind of resource on the Web created out of the system environment may be considered adequate to complement the explanation of a given topic. When a Web resource is identified as holding this condition, a teacher places it as an element of the class *awo:e-Support* and recommends it to be used by the *awo:isRecommendedBy* relation. He also cites the supported topic by the *awo:supportsTo* relation, indicates to which courses it will be available by the relation *awo:isAvailableTo*, and gives the fundamental metadata describing it. This recommendation of a resource to be used as a Learning Object is a kind of highly subjective metadata, which was indicated as very valuable [10] [19]. An element of class Taxonomy defined in <http://kmr.nada.kth.se/el/ims/schemas/lom-classification> was created in order to give learning content classification by subject matter, available at <http://www.inf.ufrgs.br/~tapejara/Domain.daml>. Learning objects are classified as having the *dc:subject* property pointing to an item in such a taxonomy.

4.4 Refinement of LOM elements

Application profiles enable increasing semantic interoperability in one community, in a way that preserves full compatibility with the larger LOM context. The refinement of properties and classes of the LOM RDF binding specification [17] contributed to the application profile's definition. To begin with, Disciplines, Courses, Topics and Support material were declared as pertaining to classes that are subclasses of the *awo:LearningObject* class, which then inherit all their restrictions on properties values. To define a controlled vocabulary with a more specialized semantics, we defined the class *awo:NetworkConnection* as a sub class of *lom:TechnologyRequirement*

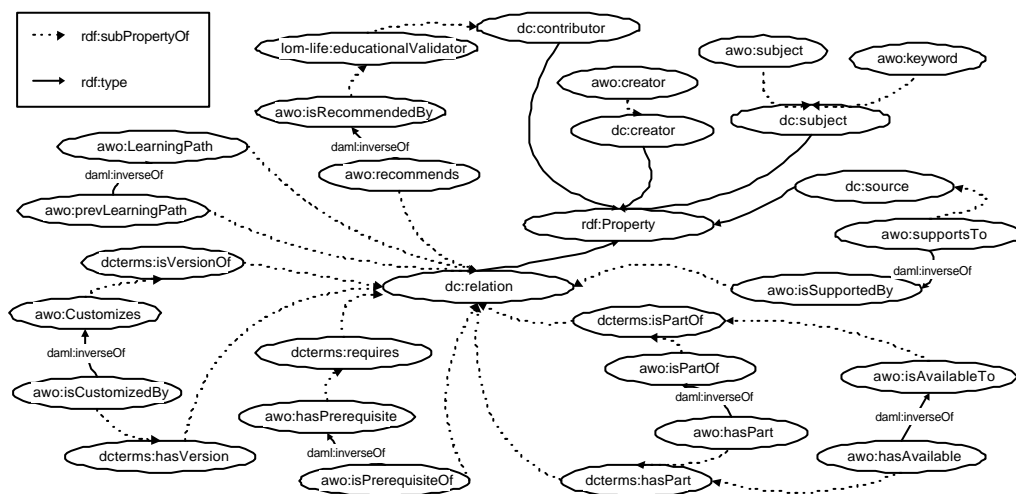


Fig. 3. Refinement of LOM properties

containing all the valid elements to be used as values of the LOM property *lom-tech:requirement*. Similarly we have defined *awo:Contributor* as a subclass of *lom:Entity* whose instances are the teachers working in the educative content creation and recommendation. Figure 3 shows, among others, the property *dcterms:requires* refinement to the *awo:hasPrerequisite* relation standing for a learning object that is needed to correctly understand the current learning object. The *awo:hasPrerequisite* relation was also declared as transitive in the ontology code, and has an inverse relation *awo:isPrerequisiteOf*, both of which are useful to enable inference tasks. Any Web agent that can not understand our ontology specifications about the *awo:hasPrerequisite* relation, can still understand the more general property *dcterms:requires*, and interpret the *awo:hasPrerequisite* relation with the semantics of the known, more general property. Similarly other specializations were defined as shown in figure 3 and table 2. All the properties used in the AdaptWeb metadata repository are those proposed by the RDF binding [17] or refinements of them.

Table 2. LOM properties refinement in the AdaptWeb application profile

Ontology Property	Refined Property	Ontology Property Description
awo:hasPrerequisite	dcterms:requires	A topic that must be taken previously to take the topic for which the property has been stated. Its domain and range are in class <i>awo:Topic</i> . It is transitive.
awo:isPartOf	dcterms:isPartOf	The whole of which the topic is part of. It is transitive. Its domain is in class <i>awo:Topic</i> . Is transitive.
awo:isAvailableTo	dcterms:isPartOf	To which courses a Topic or Support material is available. Its domain is in class <i>awo:LearningObject</i> and its range in <i>awo:Course</i> .
awo:customizes	dcterms:isVersionOf	A Discipline customization to certain general knowledge background and learning goals. Its domain is in class <i>awo:Course</i> and its range in <i>awo:Discipline</i> . It is functional.
awo:supportsTo	dc:source	Indicates that an educative material supports a Topic exposition. Its domain is in class <i>awo:Support</i> and its range in <i>awo:Topic</i> .
awo:isRecommendedBy	lom-life:educationalValidator	The person responsible for recommendations to use an external learning object. Its domain is in class <i>awo:LearningObject</i> and its range in <i>awo:Contributor</i> .
awo:learningPath	dc:relation	The order the topics explaining an idea must be presented in according to learning purposes. Its domain and range are in class <i>awo:Topic</i> .
awo:creator	dc:creator	Entities that have created the learning object. Its domain is in class <i>awo:LearningObject</i> and its range in <i>awo:Contributor</i> .
awo:keyword	dc:subject	A keyword describing a topic in the Learning Object. Its domain is in class <i>awo:LearningObject</i> and its range in <i>awo:Keyword</i> .
awo:subject	dc:subject	A term within a taxonomy used to classify learning objects by

		subject matter. Its domain is in class <i>awo:LearningObject</i> and its range in <i>lom-cls:Taxonomy</i> .
--	--	---

5 Metadata Generation

The AdaptWeb metadata repository is an ontology containing explicit, structural, contextual, and semantic metadata implementing the knowledge space model of the system. Such an ontology was encoded in DAML+OIL language [21] [22] [23] using the editor OilEd [24] to construct the model and the reasoner RACER [25] to check the model satisfiability. Metadata describing each piece of the learning content that populates the Hyperspace in terms of the knowledge space makes the necessary connection between the two spaces to let the system “know” what is presented on each piece of learning content and how is it related to each other. Such metadata is automatically generated. Each time new content is authored into the system, a PHP wrapper reads the XML files created by the authoring environment while being aware of their creation context (e.g. the metadata that can be inferred from the author information, like his or her language) and the structure and semantics of their content (i.e. the semantics inferable from the XML document tree structure and sequence of elements) generating RDF statements containing the metadata descriptions. Such metadata descriptions are Domain Knowledge ontology instances available at <http://www.inf.ufrgs.br/~tapejara/Ontology/Generated/AWOntology.daml>. At any time, the metadata repository, i.e. the ontology, can be augmented by human agents using the ontology editor OilEd [24], these augments can be over any existent element by editing comments or attribute values, and also by creating new RDF descriptions, for example pointing to resources discovered on the Web and considered valuable to complement some topic explanation. This is a way to enrich the educative content of the AdaptWeb system with resources not authored into it, but recommended to be used by identified contributors responsible for the quality of the educative content. These Web resources do not need to have previous metadata annotations describing them. The result of the enrichment by human agents is stored in <http://www.inf.ufrgs.br/~tapejara/Ontology/Augmented/AWOntology.daml>.

6 Student Model

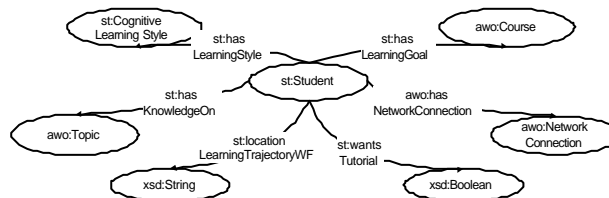


Fig. 4. Student Model

The student model graphically depicted in figure 4 consists of the central class *st:Student* with properties characterizing the student profile. The property *st:hasLearningStyle* indicates the student's cognitive style of learning (CSL) [5][6]. The property *st:hasLearningGoal* points to an element of *awo:Course* which customizes a discipline content directed to the background and objectives of a specific group of students, as defined in the Domain Knowledge ontology. The boolean property *st:wantsTutorial* indicates if the student currently prefers to work in a tutored mode, according to his hyperspace experience. Using a simple overlay model, the student's knowledge on each topic of the Knowledge Model is indicated by the relation *st:hasKnowledgeOn*, i.e. if an instance of the relation *st:hasKnowledgeOn* exists relating the student to an instance of the class *awo:Topic*, then the system believes the student has knowledge about this topic. The property *awo:hasNetworkConnection* indicates the kind of Network Connection the student has declared in the current session. The property *st:locationLearningTrajectoryWF* indicates the URL where the current learning trajectory for the student is. The remaining classes in the model, which are the range of the described properties, are as follows: *awo:Course*, *awo:Topic* and *awo:NetworkConnection* all defined in the Knowledge Space Model; *xsd:String* and *xsd:Boolean* are defined in the XML Schema namespace [15]; and *st:CognitiveLearningStyle* is defined in the namespace of the Student Model: <http://www.inf.ufrgs.br/~tapejara/Ontology/Student.daml>, containing the four CLS identified in [5], i.e. (i) Analogue-Analytical; (ii) Concrete-Generic; (iii) Deductive-Evaluative and (iv) Relational-Synthetic. According to his CLS, a student may prefer to visit the examples about a topic, try to do the exercises, and then go on to the page containing the abstract explanation of the topic, while another student with a different CSL may prefer to read the abstract explanation of the topic content first. From this we can observe that the tutorial workflow path proposed by the system for each student will be strongly influenced by his CLS. The student model was implemented as an ontology encoded in DAML+OIL language [21] [22] [23] using the editor OilEd [26] to construct the model and the reasoner RACER [25] to check the model satisfiability. Its instances are created or updated automatically at the beginning of each session with the information the student gives about himself and also by the result of the continuous analysis of the student's activities [5].

7 Adaptive Content Selection.

Since ontology relations can be viewed as binary predicates in first order logic and class membership as unary predicates, we can use first order logic to represent statements about ontology elements. As an example scenario to show the adaptive content selection, we can assume a student *Stud1* graduated in Computer Science, with a dial-up network connection that intends to take a Numerical Methods course, then `learningGoal(Stud1, Computer Science)` is stated in the ontology because at authoring time it was indicated that *ComputerScience* is a course of Numerical Methods

discipline customized to Computer Science students, and *learningGoal* is an ontology property relating students with learning goals. Also *networkConnection(Stud1, "dial-up")* is stated indicating the kind of connection that the student currently has. To begin with, it is necessary to determine what the first content to be offered to the student is. The educative content related to topics x available for course *Computer Science* with no prerequisites that also begin a learning path, are the learning objects the student is able to take at this stage. The following is computed.

$$\text{hasAvailable}(\text{Computer Science}, x) \wedge \neg \exists y(\text{hasPrerequisite}(x, y)) \wedge \neg \exists z(\text{learningPath}(z, x))$$

Also, the student's Cognitive Style of Learning (CLS) is taken into account to define the first element among those selected above to give to the student. The existence of the following assertions indicating that the student has the CLS "Analogue-Analytical", and the order the students with CLS "Analogue-Analytical" usually prefer the educative material is assumed.

$$\begin{aligned} &\text{hasLearningStyle}(\text{Stud1}, \text{"Analogue-Analytical"}); \\ &\text{first}(\text{Analogue-Analytical}, z) \rightarrow \text{Example}(z); \\ &\text{second}(\text{Analogue-Analytical}, t) \rightarrow \text{Topic}(t) \end{aligned}$$

The sentences above indicate that students with CLS Analogue-Analytical prefer to take a simple example prior to taking the theoretical exposition of a new topic. According to what is stated above, it will be concluded that the first learning object to offer the student must be an example s of Topic x whose degree of difficulty is *easy*. Additionally it must be a version suitable to the speed of the student network connection. Then, the following must hold:

$$\text{isSupportedBy}(x, s) \wedge \text{Example}(s) \wedge \text{requirement}(s, \text{"dial-up"}) \wedge \text{difficulty}(s, \text{"Easy"})$$

It can be observed that the inverses of relations *isAvailableTo* and *supportsTo* were used to compute the educative content available for the course and the support material available for a given topic respectively.

8 System Architecture

The proposed system architecture is depicted in figure 5. At the content creation time, the **Authoring** module provides a methodology and an editing tool to create new educative content. During the authoring process, the author can consult the Domain Knowledge ontology to be aware of pre-existing learning objects to eventually reuse them. Each time new content is created, the **Automatic Metadata Generation** wrapper generates its basic metadata according to what was explained in section 5, and store it in the Domain Knowledge ontology. At any time the **Metadata Augmentation** module

can enable the Domain Knowledge ontology augmentation by human agents through the use of the editor OilEd [24]. A possible use of this metadata augmentation feasibility is to create metadata descriptions pointing to resources discovered on the Web and recommended by a teacher as adequate to complement some topic explanation. The *Student Monitoring* agent continuously updates the *Student Model* according to the activities of the student. The *Adaptive Content Selection* agent defines a teaching strategy to guide dynamic content sequence generation. It selects the specific contents to be presented, creating a learning trajectory tailored to the student profile. The *Adaptive Presentation* module determines the presentation style according to the preferences stated in the student model using the strategy of disabling links if the student has chosen the tutorial mode for his navigational preferences and offers the search by keywords if he has decided to learn in a free navigational style.

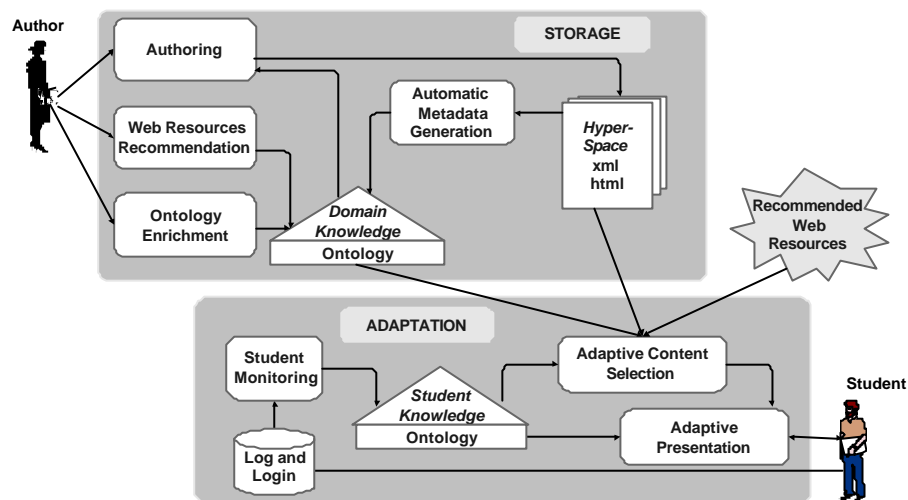


Fig. 5. System Architecture

9 Conclusions and Future Work

This paper describes the steps taken in the design and implementation of an in – progress project for adaptive content presentation of Web-based courses delving into the definition of an application profile of the Learning Object Metadata (LOM) standard [9] to represent the metadata of their educational content and the design and implementation of the student and domain model as Web ontology. The proposed solution is demonstrated with the application of the AdaptWeb system. A use case was implemented with the creation of the metadata concerning a Numerical Computation course, where specific and different contents of the same discipline are presented for students of three different programs (Mathematics, Engineering, or Computer Sci-

ence). In the adaptive environment provided by the system, course contents are customized with different information complexity, sequence of contents access, example application, and supplementary material access. Authoring software is provided for syllabus generation, supported by XML standard files. The course content is adapted and presented to students, according to each student's program, current knowledge, and navigation preferences. An identified future work is to relate the top of each taxonomy used to classify learning objects by subject matter in the knowledge Space Model with the criteria of Discipline or Idea with a more general taxonomy existing on the Web, as the ACM Computer Classification system [33]. Additionally, future versions of the ontology will be encoded in OWL DL language [26].

References

- [1] Brusilovsky, P. *Methods and Techniques of Adaptive Hypermedia*. Adaptive Hypertext and Hypermedia. Kluwer Academic Publishers, pp. 1-43, 1998.
- [2] Brusilovsky, P. *Adaptive and Intelligent Technologies for Web-based Education*. In C. Rollinger and C. Peylo editors, *Special Issue on Intelligent Systems and Teleteaching*, *Kunstliche Intelligenz*, 4, 19-25, 1999.
- [3] Rousseau, F. Garcia-Macías, De Lima, J. Duda, A. *User Adaptable Multimedia Presentations for the WWW*. WWW1999, May 11-14, 1999, Toronto, Canada.
- [4] Brusilovsky, P. *Developing Adaptive Educational Hypermedia Systems: From Design Models to Authoring Tools*. in T. Murray, S. Blessing and S. Ainsworth (Eds.), Ablex, Norwood, 2002.
- [5] Souto, M. Nicolao, M. Viccari, R. Palazzo, J. Verdin, R. Beschoren, K. Madeira, M. Zanella, R. *Web Adaptive Training System Based on Cognitive Student Style*. IFIP World Computer Congress, 17th Edition, August 25-30, 2002, Montreal, Canada.
- [6] Souto, M. Verdin, R. Wainer, R. Madeira, M. Wapechowsky, M. Beschoren, K. Zanella, R. Correa, J. Vicari, R. Palazzo, J. *Towards an Adaptive Web Training Environment Based on Cognitive Style of Learning: an Empirical Approach*. AH 2002, 2002, Malaga, Spain.
- [7] Palazzo M. de Oliveira, J.; Silva Muñoz, L.; de Freitas, V. Marçal, V; Gasparini, I.; Abrahão Amaral, M. *AdaptWeb: an Adaptive Web-based Courseware* the 3rd Annual Ariadne Conference, 20-21 November 2003, Belgium.
- [8] Gruber, T. R. *A translation approach to portable ontology specifications*. *Knowledge Acquisition*, 5:199–220 1993.
- [9] *Learning Technology Standards Committee of the IEEE. Draft Standard for Learning Object Metadata (LOM) IEEE 1484.12.1-2002*.
- [10] Duval, E. Hodgins, W. Sutton, S. Weibel, S. *Metadata Principles and Practicalities*. *D-Lib Magazine* 8(4), 2002.
- [11] Heery, R. Patel, M. *Application profiles: mixing and matching metadata schemas*. September 2000, ARIADNE Issue 25.

- [12] The Dublin Core Metadata initiative. <http://dublincore.org/>
- [13] Bray, T. Paoli, J. Sperberg-McQueen, C. Maler, E. Extensible Markup Language (XML) 1.0 (Second Edition) W3C Recommendation, October 6, 2000.
- [14] Lassila, O. Swick, R. Resource Description Framework (RDF) model and syntax specification. W3C Recommendation, 1999.
- [15] Fallside, D. XML Schema Part 0: Primer W3C Recommendation, 2 May 2001.
- [16] Brickley, D. Guha, R. RDF Vocabulary Description Language 1.0: RDF Schema. W3C Working Draft 10 October 2003.
- [17] Nilsson, M. IEEE Learning Object Metadata RDF binding. May 2001. <http://kmr.nada.kth.se/el/ims/metadata.html>.
- [18] Vcard schema. <http://www.w3.org/2001/vcard-rdf/3.0>.
- [19] Duval, E. Hodgins, W. A LOM Research Agenda. WWW2003, May 20-24, 2003, Budapest, Hungary.
- [20] Namespaces in XML, World Wide Web Consortium, 14-January 1999, Editors: Tim Bray, Dave Hollander and Andrew Layman. <http://www.w3c.org/TR/REC-xml-names>.
- [21] Connolly, D. van Harmelen, F. Horrocks, I. McGuinness, D. Patel-Schneider, P. Stein, L. DAML+OIL (March 2001) Reference Description. W3C Note 18 December 2001.
- [22] Connolly, D. van Harmelen, F. Horrocks, I. McGuinness, D. Patel-Schneider, P. Stein, L. Annotated DAML+OIL Ontology Markup. W3C Note 18 December 2001.
- [23] Horrocks, I. DAML+OIL: a Description Logic for the Semantic Web. IEEE Data Engineering Bulletin 25(1): 4-9 (2002).
- [24] Bechhofer, S. OilEd 3.5 Manual. Information Management Group, Department of Computer Science, University of Manchester. <http://www.cs.man.ac.uk/~horrocks/software>
- [25] Haarslev, V.; Moller, R. RACER User's Guide and Reference Manual Version 1.7.7. September 17, 2003. <http://www.fh-wedel.de/~mo/racer/index.html>.
- [26] Bechhofer, S. van Harmelen, F. Hendler, J. Horrocks, I. McGuinness, D. Patel-Schneider, P. Stein, L. OWL Web Ontology Language Reference. W3C Note 18 August 2003.
- [27] Ariadne. <http://www.ariadne-eu.org>.
- [28] Heal. <http://www.healcentral.org>.
- [29] Cancore. <http://www.cancore.org>.
- [30] Merlot. <http://www.merlot.org>.
- [31] SingCore. <http://www.ecc.org.sg/eLearn/MetaData/SingCORE/index.jsp>.
- [32] NILSSON, Mikael. The semantic web: How RDF will change learning technology standards. CETIS, September 27, 2001 <http://www.cetis.ac.uk/content/20010927172953>.
- [33] The ACM Computing Classification System. <http://www.acm.org/class/1998>.