Dynamic Assembly of Personalized Learning Content on the Semantic Web

Jelena Jovanović[1], Dragan Gašević[2], and Vladan Devedžić[1]

[1] FON, School of Business Administration, University of Belgrade, Serbia and Montenegro
{jeljov, devedzic}@fon.bg.ac.yu
[2] School of Interactive arts and Technology, Simon Fraser University Surrey Canada
dgasevic@sfu.ca

Abstract. This paper presents an ontology-based approach for automatic decomposition of learning objects (LOs) into reusable content units, and dynamic reassembly of such units into personalized learning content. To test our approach we developed TANGRAM, an integrated learning environment for the domain of Intelligent Information Systems. Relying on a number of ontologies, TANGRAM allows decomposition of LOs into smaller content units, which can be later assembled into new LOs personalized to the user's domain knowledge, preferences, and learning styles. The focus of the presentation is on the ontologies themselves, in the context of user modeling and personalization. Furthermore, the paper presents the algorithm we apply to dynamically assemble content units into personalized learning content. We also discuss our experiences with dynamic content generation and point out directions for future work.

1 Introduction

Reusing learning objects (LOs) across educational applications is a great idea, but not easily achievable in practice. A recent study by Brooks et al. [1] has shown that current e-learning standards and specifications (such as the IEEE LOM standard) are rather restrictive in terms of the variety of metadata they capture and imprecise in expressing the structure of such metadata. Moreover, few of the metadata fields proposed by such specifications are actually used in learning object repositories (LORs) to annotate the LOs, which reduces the possibility for agents to retrieve the LOs. As a result, nearly all LO-based courses are created directly by instructional designers, who explicitly hand craft the LOs for the purpose. Furthermore, Robert and Gingras [12] conducted an experiment showing that teachers mostly reuse their own material, and only some LOs created by other teachers. The reusability of other people's LOs largely depends on the teacher's instructional practices and teaching style, as well as on the type of content of those LOs (presentations, diagrams, tests, etc.). The practice of handcrafting new LOs from existing ones shows that authors very often copy-and-paste parts of existing LOs into newly created LOs. In other words, rather than reusing entire LOs for their courses, they manually reuse their parts.

This creates the idea of reusable content units at a granularity finer than LO as a whole. We have developed an ontology-based approach for automatic decomposition

Y. Sure and J. Domingue (Eds.): ESWC 2006, LNCS 4011, pp. 545 – 559, 2006.

of LOs into reusable fragments, and dynamic reassembly of such fragments into personalized learning content.

1.1 Problem Statement

The objectives of this paper are:

- to explain the rationale for using ontologies to enable on-the-fly assembly of personalized learning content out of reusable content units;
- to present an example of how such an ontology-based approach is implemented in a specific learning environment, called TANGRAM;
- to discuss practical implementation details and experience with dynamic generation of personalized learning content.

The focus of the presentation is on the ontologies themselves, in the context of user modeling and personalization. The principles we discuss are implementation-independent. On the other hand, their implementation in TANGRAM helped us reveal important practical details we were not aware of initially.

The rest of the paper is structured to follow the order of the objectives stated above.

2 The Rationale

The approach that we propose can be summarized as follows: reuse existing content units to dynamically generate new learning content tailored to satisfy the needs of a specific student. To overcome the problem of interoperability between disparate domains, we based our approach on Semantic Web technologies, ontologies in particular.

The starting point in our approach is the classification of ontologies in the domain of eLearning suggested in [13]. This classification differentiates between the following types of ontologies: 1) content (domain) ontologies that formally describe the subject matter (topics) of learning content; 2) structural ontologies that formalize the content structure; and 3) context ontologies that specify the pedagogical/instructional role of the content. In our approach, a LO is represented in a structural ontology compliant format, whereas concepts of a domain ontology are used to semantically describe the LO's content. In addition, the concepts from a context ontology are used to mark up LOs with their pedagogical/instructional roles. The proposed approach also assumes annotation of each component of a LO, thus making individual components reusable.

Explicitly defined structure of a LO facilitates adaptation of the LO, as it enables direct access to each of its components and their tailoring to the specific features of a student. Besides, being able to directly access components of a LO, we are empowered to dynamically, on-the-fly create new, personalized learning content.

To be reusable, a domain ontology must not contain any information related to topics sequencing and navigation. On the other hand, it does make sense to formally represent an optimal learning path through the domain. Accordingly, we use a special ontology for that purpose. Finally, a user model ontology is used to enable formal representation of users' data and exchange of these data with other learning applications.

3 Ontologies for Dynamic Assembly of Personalized Content

To test the feasibility of the proposed approach to dynamic assembly of personalized learning content, we have developed TANGRAM – an integrated learning environment for the domain of Intelligent Information Systems (IIS). TANGRAM is implemented as a Web application built on top of a repository of educational content and intended to be useful to both content authors and students interested in the domain of IIS. Fig. 1 illustrates TANGRAM's architecture and depicts the ontologies it uses. These ontologies are concisely described in the following subsections[1]. Additionally, to annotate content units in TANGRAM, we defined a profile of the IEEE LOM RDF Binding specification[2]. The profile defines a subset of the IEEE LOM elements that we found necessary to support the intended functionalities of the system [9].

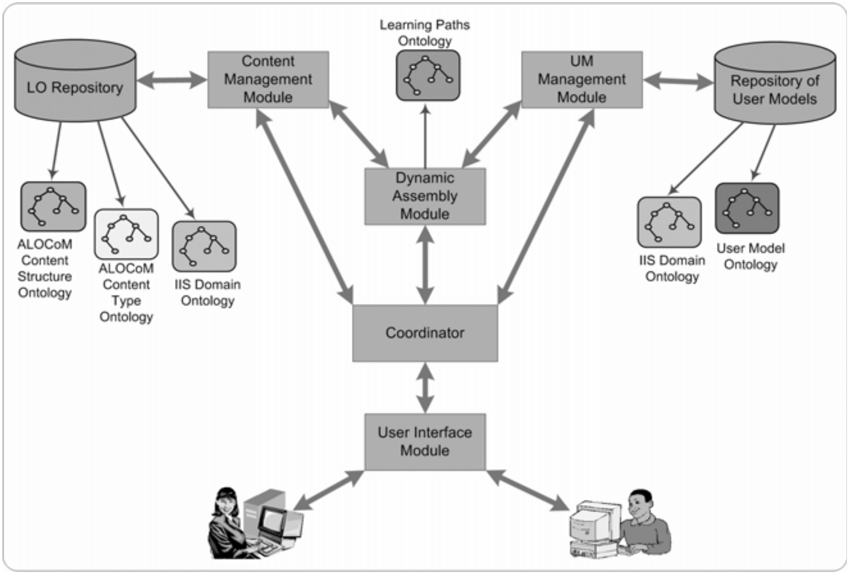


Fig. 1. TANGRAM's architecture

3.1 ALOCoM-Based Ontologies

In our previous collaborative research efforts with the ARIADNE research group from K.U. Leuven, Belgium, we developed ALOCoM ontology as a content structure ontology based on the Abstract Learning Object Content Model (ALOCoM) [14]. The ontology defines concepts and relationships that enable formal definition of the structure of a LO. To learn more about this ontology, interested readers should refer to [10]. However, our latest research led to a major revision of the ALOCoM ontology and its division into: ALOCoM Content Structure ontology (ALOCoMCS) and ALOCoM Content Type ontology (ALOCoMCT).

[1] All ontologies can be downloaded from: http://iis.fon.bg.ac.yu/TANGRAM/ ontologies.html

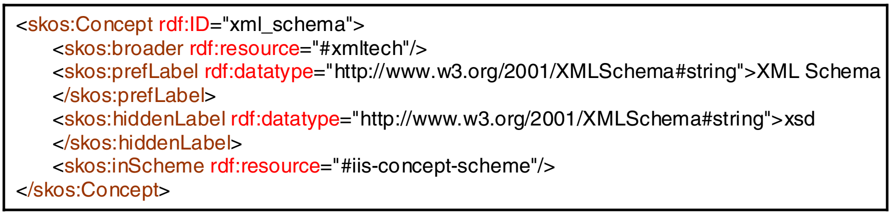
[2] http://kmr.nada.kth.se/el/ims/md-lomrdf.html

Being based on the common model, these two ontologies share the same root concepts: Content Unit (CU), Content Fragment (CF), Content Object (CO) and Learning Object (LO). CU is an abstract concept aimed at representing content of any level of granularity. CFs are CUs in their most basic form, like text, audio and video. These elements can be regarded as raw digital resources that cannot be further decomposed. A CO is an aggregation of CFs and/or other COs. Navigational elements enable sequencing of CFs in a CO. LOs aggregate COs around a learning objective. However, in our ALOCoM-based ontologies, these basic types of CUs are considered from completely different perspectives – ALOCoMCS is about content structuring, whereas ALOCoMCT focuses on potential instructional/pedagogical roles of CUs.

3.2 Domain Ontology

The SKOS Core ontology[3] is used as the basis of the IIS course domain ontology[4]. Being specifically developed to describe taxonomies and classification schemes, the SKOS Core ontology has an excellent variety of properties to describe relationships between topics in a course.

Each concept of the IIS domain is represented as an instance of the *skos:Concept* class, whereas the conceptual scheme of the domain is represented as an instance of the *skos:ConceptScheme* class. The SKOS' property *skos:inScheme* is used to associate all defined instances of the *skos:Concept* class to the conceptual scheme of the IIS domain. Likewise, each identified domain concept is assigned one or more aliases (i.e., alternative terms typically used in literature when referring to a concept) using the SKOS properties *skos:prefLabel*, *skos:altLabel*, and *skos:hiddenLabel*. SKOS semantic properties, i.e. properties derived from the *skos:semanticRelation* property, enabled us to structure the IIS domain in a generalization hierarchy (via the *skos:broader* and its inverse *skos:narrower* properties), as well as to define semantic relations between concepts belonging to different branches of the hierarchy (via the *skos:related* property). We used the *skos:hasTopConcept* property to relate the most general domain concepts (such as intelligent agents, Semantic Web, etc.) to the IIS concept scheme, thus formally stating that these concepts form the top level of the created concepts hierarchy. Fig. 2 shows an excerpt of the ontology that defines 'XML Schema' as a domain concept.



```
<skos:Concept rdf:ID="xml_schema">
     <skos:broader rdf:resource="#xmltech"/>
     <skos:prefLabel rdf:datatype="http://www.w3.org/2001/XMLSchema#string">XML Schema
     </skos:prefLabel>
     <skos:hiddenLabel rdf:datatype="http://www.w3.org/2001/XMLSchema#string">xsd
     </skos:hiddenLabel>
     <skos:inScheme rdf:resource="#iis-concept-scheme"/>
</skos:Concept>
```

Fig. 2. excerpt from the SKOS-based IIS domain ontology

[3] http://www.w3.org/2004/02/skos/core/

[4] Actually, we used SKOS Core OWL binding available at: http://ai.usask.ca/mums/schemas/2005/01/27/skos-core-dl.owl

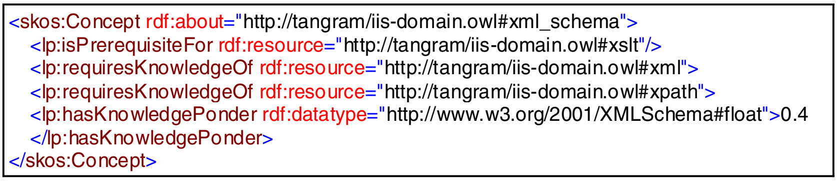
One should note that the domain ontology does not contain any information regarding topics sequencing, in terms of the order in which the topics should be presented to the learners. That kind of information is stored separately in the Learning Paths ontology.

3.3 Learning Paths Ontology

The Learning Paths (LP) ontology defines learning trajectories through the topics defined in the domain ontology. We defined this ontology as an extension of the SKOS Core ontology that introduces three new properties: *lp:requiresKnowledgeOf*, *lp:isPrerequisiteFor*, and *lp:hasKnowledgePonder*. The first two are semantic properties defining prerequisite relationships between domain topics, whereas the third one defines difficulty level of a topic on the scale from 0 to 1.

The properties *lp:requiresKnowledgeOf* and *lp:isPrerequisteFor* are defined as sub-properties of the *skos:semanticRelation* property of the SKOS Core ontology. These properties are defined as mutually inverse and transitive. One should note that unlike the Dublin Core properties *dc:requires* and *dc:isRequiredBy*[5] that establish dependency of prerequisite type among physical LOs, the properties we introduced are intended to describe similar relations on the level of domain concepts.

As Fig. 3 suggests, the LP ontology relates instances of the domain ontology through an additional set of relationships reflecting a specific instructional approach to teaching/learning IIS. The main benefit of decoupling the domain model in this way is to enable reuse of the domain ontology – even if the applied instructional approach changes, the domain ontology remains intact.



```
<skos:Concept rdf:about="http://tangram/iis-domain.owl#xml_schema">
   <lp:isPrerequisiteFor rdf:resource="http://tangram/iis-domain.owl#xslt"/>
   <lp:requiresKnowledgeOf rdf:resource="http://tangram/iis-domain.owl#xml">
   <lp:requiresKnowledgeOf rdf:resource="http://tangram/iis-domain.owl#xpath">
   <lp:hasKnowledgePonder rdf:datatype="http://www.w3.org/2001/XMLSchema#float">0.4
   </lp:hasKnowledgePonder>
</skos:Concept>
```

Fig. 3. An excerpt from the Learning Paths ontology for the domain of IIS

3.4 User Model Ontology

We developed a User Model (UM) ontology to help us formally represent relevant information about TANGRAM users (content authors and students). The ontology focuses exclusively on the user information that proved to be essential for TANGRAM's functionalities. To enable interoperability with other learning applications and exchange of users' data, we based the ontology on official specifications for user modeling: IEEE PAPI Learner[6] and IMS LIP[7]. Furthermore, since we did not want to end up with another specific interpretation of the official specifications, potentially incompatible with existing learning applications, we explored existing solutions, like the ones presented in [4] and [11]. The result is a modular UM ontology that:

[5] http://dublincore.org/documents/dcmi-terms/

[6] http://edutool.com/papi

[7] http://www.imsglobal.org/profiles

- uses some parts of the UM ontology developed for the ELENA project and described in [4]; specifically, we use the elements aimed for representing students' performance (as proposed by the IEEE PAPI Learner specification) and their preferences (as specified in the IMS LIP);
- introduces new constructs for representing users' data that the official specifications do not declare and the existing ontologies either do not include at all, or do not represent in a manner compliant to the needs of TANGRAM.

In the center of Fig. 4[8] one can notice class *um:User* that formally describes the concept of a TANGRAM user. Each user, i.e. instance of this class, is related to a set of his/her personal data via the *um:hasPersonalInfo* property. Personal data are formally represented with the *um:PersonalInfo* class and its datatype properties: *um:username* and *um:password* properties that keep values of secure login data, as well as *um:name* property representing the user's name. Each user can be a member of one or more organizations (*um:Organization*). Specifically, the user can be a member of a university (*um:University*), a research centre (*um:ResearchCentre*) and/or a research group (*um:ResearchGroup*). Additionally, for each user the system needs data about his/her role/position in the formal organization (s)he belongs to. Therefore,

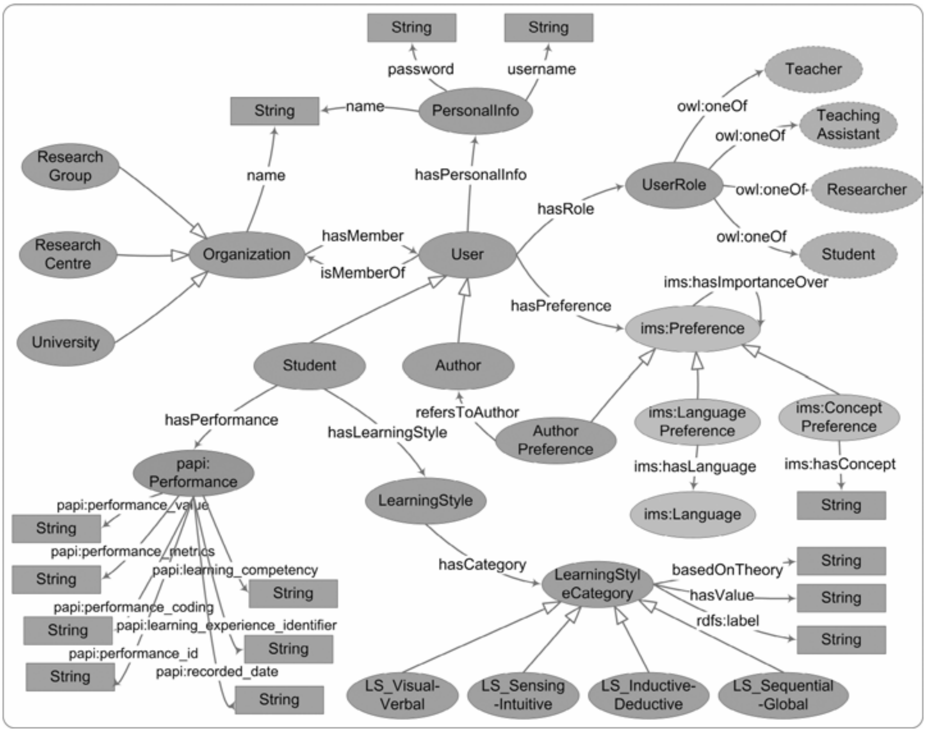


Fig. 4. Graphical representation of the TANGRAM's User Model Ontology

[8] Classes and properties that do not have namespace prefix in Fig. 4 belong to the um: http://tangram/user-model/complete.owl namespace.

we introduced property *um:hasRole* that relates an instance of the *um:User* class with an appropriate instance of the *um:UserRole* class. The latter class formalizes the concept of a role/position a user typically has in an educational environment and is specified as an enumeration (via *owl:oneOf* construct) of the following instances: *um:Teacher*, *um:TeachingAssistant*, *um:Researcher*, *um:Student*. Of course, this enumeration can be extended to encompass additional roles if needed. Further, each user can have certain preferences (*um:hasPreference*) regarding language (*ims:-LanguagePreference*) and/or domain topics (*ims:ConceptPreference*). Representation of users' preferences is taken from the user model ontology developed for the ELENA project [4] and is fully compliant with the IMS LIP specification (hence *ims* prefix). Class *ims:Preference*, formally representing a user's preference, can have *ims:hasImportanceOver* property that defines priority (i.e. importance) of a preference for a specific user. Furthermore, the ontology introduces *um:AuthorPreference* class as a subclass of *ims:Preference* in order to represent users' preferences regarding authors of learning content. The property *um:refersToAuthor* associates this specific type of a user's preference with his/her favorite author of learning content (one or more of them).

The remaining classes and properties of the TANGRAM UM ontology are exclusively aimed at formal representation of students' data. Each student (*um:Student*) is assigned a set of performance-related data (via *um:hasPerformance* property) represented in the form of the *papi:Performance* class and the following set of properties[9]:

1. the *papi:learning _competency* property refers to a concept of the domain ontology that formally describes the subject matter of the acquired knowledge in the best way (i.e. contains URI of that concept);
2. the *papi:learning_experience_identifier* property identifies a CU that was a part of the learning material used for learning. In TANGRAM, each instance of the *papi:Performance* class has a number of properties of this type – one for each CU used to assemble the learning content for the student;
3. the *papi:performance_coding* and *papi:performance_metrics* properties define respectively the coding system and the metrics used to evaluate a student's performance level (i.e., the level of the acquired knowledge);
4. the *papi:performance_value* property keeps information about the real value/level of the acquired knowledge measured in terms of the specified metrics and coding system;
5. the *papi:recorded_date* property is aimed at representing date and time when the performance was recorded, i.e. when the learning process took place.

Additionally, for each student the system keeps data about his/her learning style. Representation of learning styles in the UM ontology is based on the Felder & Silverman model of learning styles [6]. This model recognizes 5 categories of learning styles: 1) Visual-Verbal, 2) Sensing-Intuitive, 3) Sequential-Global, 4) Inductive-Deductive and 5) Active-Reflective. The learning style of a student is formally represented by the *um:LearningStyle* class in the UM ontology. This class is associated (via the *um:hasCategory* property) with the *um:LearningStyleCategory* class that formally stands for one specific aspect (category) of the learning style. Specifically, TANGRAM implements the learning categories defined in the Felder & Silverman

[9] The prefix *papi:* is used to denote that the Performance class and its properties are defined according to the PAPI Learner Specification.

model and introduces one subclass of the *um:LearningStyleCategory* class to represent each of those categories (e.g. *um:LS_Visual-Verbal*)[10]. To make the ontology more general and easily extensible, we assigned the property *um:basedOnTheory* to the *um:LearningStyleCategory* class, thus enabling the introduction of learning style categories defined by other authors. The class *um:LearningStyleCategory* is also attached the *um:hasValue* property aimed at representing the position of a specific student on the continuum defined by the opposite poles of a learning style category. The range of this property is restricted to double values between -1 and 1 (inclusively). The boundary values (-1 and 1) represent the two extreme poles of each learning style category. For example, assigning the value of -1 to the *um:hasValue* property of the *um:LS_Visual-Verbal* class means that the learner is highly visual. On the opposite, *um:hasValue* property with the value of 1 identifies a highly verbal learner.

4 Personalized Learning in TANGRAM

TANGRAM provides adaptation of learning content to the specific needs of individual students. Currently, it is focused on enabling personalized learning experience to students interested in the domain of IIS. Two basic functionalities of the system from the students' perspective are:

- Provision of learning content adapted to the student's current level of knowledge of the domain concept of interest, his/her learning style, and other personal preferences.
- Quick access to a particular type of content about a topic of interest, e.g. access to *example*s of RDF documents or *definition*s of the Semantic Web (both topics belong to the domain of IIS).

In this section we focus on the former functionality and explain in details how it is implemented in TANGRAM.

4.1 Initialization of the User Model

A student must register with the system during the first session. Through the registration procedure the system acquires information about the student sufficient to create an initial version of his/her model. The student is required to fill in a simplified version of the Felder&Silverman questionnaire for determining the student's learning style[11]. The acquired data enables the system to create personalized learning content for the student.

As for initial determination of the student's knowledge about the IIS domain, the system relies on the student's self-assessment. During the registration procedure, the student is asked to estimate his/her level of knowledge of the main sub-domains of the IIS domain (e.g. Intelligent Agents, Semantic Web). In particular, the student is presented with the following set of options: 'Never heard of the topic', 'Have a basic idea', 'Familiar with', 'Know well' and 'Demand advanced topics', and has to choose

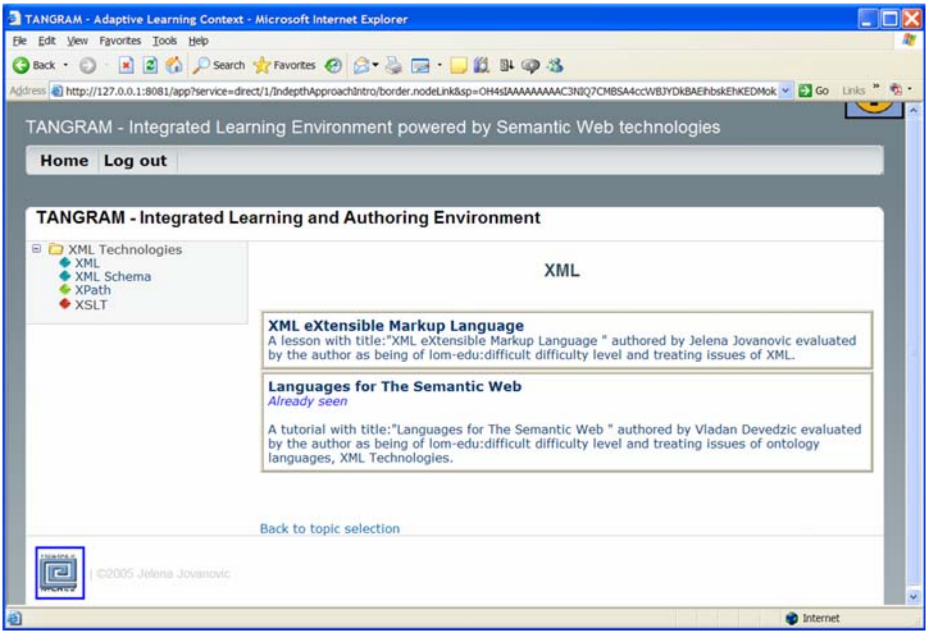
[10] We did not consider Active-Reflective learning style category, as it emphasizes social aspects of a learning process that TANGRAM currently is not able to support.

[11] The questionnaire is known as "Index of Learning Styles", and is available at http://www.engr.ncsu.edu/learningstyles/ilsweb.html

the one that reflects his\her knowledge best. Internally, TANGRAM converts the student's selection for each sub-domain into its numerical counterpart (0, 0.2, 0.4, 0.6 or 0.8, respectively). These numerical values are later compared to the values of the *lp:hasKnowledgePonder* property assigned to the domain concepts in the LP ontology, to let the system determine the student's initial position in the IIS domain space and provide him/her with proper guidance and support.

4.2 Dynamic Assembly of Personalized Learning Content

A learning session starts after the user (registered and authenticated as a student) selects a sub-domain of IIS to learn about. The system performs a sort of comparative analysis of data stored in the student's model and in the LP ontology. Specifically, the LP ontology is queried for the set of domain concepts that are essential for successful comprehension of the topics from the chosen sub-domain. More precisely, the query targets the concepts related via *lp:requiresKnowledgeOf* property to the topics encompassed by the chosen sub-domain. Subsequently, the student model is queried for data about the student's level of knowledge about the selected sub-domain and the identified set of prerequisite concepts. Information resulting from this analysis is used to provide adaptive guidance and direct the student towards the most appropriate topics for him/her at that moment. To achieve this, we make use of link annotation and hiding techniques [2]. Specifically, hierarchical organization of concepts of the selected sub-domain is visualized as an annotated tree of links (shown in the upper left corner of Fig. 5). We use the following link annotations:


TANGRAM - Adaptive Learning Context - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
Go
Links
Internet

http://127.0.0.1:8081/app?service=direct/1/IndepthApproachIntro/border.nodeLink&sp=OH4sIAAAAAAAAC3NIQ7CMBSA4ccWBJYDkBAEhbskEhKEDMok

TANGRAM - Integrated Learning Environment powered by Semantic Web technologies

Home Log out

TANGRAM - Integrated Learning and Authoring Environment

- XML Technologies
  - XML
  - XML Schema
  - XPath
  - XSLT

XML

XML eXtensible Markup Language
A lesson with title:"XML eXtensible Markup Language " authored by Jelena Jovanovic evaluated by the author as being of lom-edu:difficult difficulty level and treating issues of XML.

Languages for The Semantic Web
Already seen

A tutorial with title:"Languages for The Semantic Web " authored by Vladan Devedzic evaluated by the author as being of lom-edu:difficult difficulty level and treating issues of ontology languages, XML Technologies.

Back to topic selection

©2005 Jelena Jovanovic

Fig. 5. Screen shot of a page presenting a ranked list of generated assemblies (i.e. their descriptions)

1. blue bullet preceding a link to a domain concept denotes that the student knows the topic that the link points to,
2. green bullet denotes a recommended domain concept, i.e. a concept that the student has not learned yet, but has knowledge about all prerequisite topics,
3. red bullet is used to annotate a domain topic that the student is still not ready for as (s)he is ignorant of the prerequisite topics.

Link hiding technique is used to prevent the student from accessing topics that are too advanced for him/her. In other words, links annotated with red bullets are made inactive.

After the student selects one concept from the topics tree, the system initiates the process of dynamic assembly of learning content on the selected topic. The process is based on the following algorithm:

1. *Query the LOR for content units covering the selected domain topic.* The query is based on the *dc:subject* metadata element of the CUs from the repository. If the repository does not contain CUs on the selected topic, the further steps of the algorithm depend on the student's learning style, i.e. on its Sequential-Global dimension, to be more precise[12]. If the student belongs to the category of global learners, the algorithm proceeds normally. Otherwise, the system informs the student that the learning content on the selected topic is currently unavailable and suggests other suitable topics.
2. *Classify the retrieved content units into groups according to the same parent LO criterion.* In other words, CUs originating from the same slide presentation are put in the same group.
3. *Sort components in each group.* The sorting procedure is based on the original order of CUs from the group, i.e. on the value of the *alocomcs:ordering* property of the parent LO. In the subsequent text we use the term *assembly* to refer to a group of CUs sorted in this manner.
4. *Rank assemblies according to their compliance with the student model.* Each assembly is assigned a double value (relevancy) between 0 and 1 that reflects its compliance with the student's model, i.e. its relevancy for the student. To calculate the relevancy of an assembly we query the student's model for the data about the student's learning style, his/her preferred author as well as his/her learning history data (already seen CUs). The greater the value of the relevancy, the higher the importance of the assembly for the student.
5. *Present the student the sorted list of assemblies' descriptions* and let him/her decide which one to take (Fig. 5). Description of an assembly is actually the value of the *dc:description* metadata element attached to the LO that the content of the assembly originates from. One should note that the TANGRAM does not aim to make a choice for a student. Instead, the system provides guidance to the student (using link annotation and hiding techniques), and eventually lets him/her decide on the assembly to learn from.
6. *Show the student the learning content from the selected assembly.* As soon as the student selects one assembly from the list, the system presents its content using its generic form for presentation of dynamically assembled learning content.

[12] Whereas global learners prefer holistic approach and learn best when provided with a broader context of the topic of interest, sequential learners tend to be confused/disoriented if the topics are not presented in a linear fashion [6].

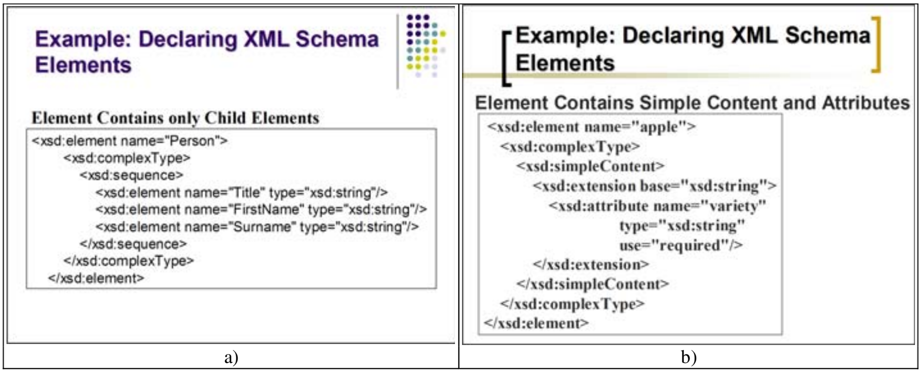
7. *Update the student model.* Specifically, the system creates an instance of the *papi:Performance* class in the student model and assigns values to its properties (see Section 3.4 for details). For example, the *papi:performance_value* property is assigned a value that reflects the level of mastery of the domain topic. If it was a topic recommended by the system, the property is assigned the maximum value (1). However, if the assembly covered an advanced topic, due to the lack of more appropriate learning content, this property is set to 0.35. This approach was inspired by the work of De Bra et al [3] and is based on the assumption that the student, due to the lack of the necessary prerequisite knowledge was not able to fully understand the presented content.

5 Discussion

In this section we discuss our experiences with the process of dynamic content assembly, emphasizing its most challenging aspects. Actually, we draw attention to the deficiencies of the presented algorithm and explain their origins.

Current implementation of the algorithm explained above uses exclusively slides (instances of *alocomc:Slide* class) for dynamic generation of personalized learning content. All our attempts to base the assembly process on CUs of lower granularity levels (*alocomcs:Paragraph*, *alocomcs:List*, *alocomcs:ListItem*,...) ended unsuccessfully: we did not manage to automatically generate coherent learning content out of those components. Additionally, one might argue that an assembly is nothing more than a slide presentation from which someone has taken out slides that do not deal with the relevant domain topic(s). However, it should be noted that our original idea was completely different. We intended to build new learning materials by combining CUs from diverse LOs. Nonetheless, this objective turned out as too ambitious: proper sequencing of small size components, as well as meaningful arrangement of their content, authoring styles, terminology and other relevant features proved to be an insurmountable task.

We recognized the lack of precise semantic descriptions of a CU's content as the major obstacle for using small-size CUs in the process of automatic content assembly. To make these statements clearer, let us consider a small example. Fig. 6 presents two slides from different slide presentations, authored by different authors, but covering the same domain concept – the concept of the XML Schema. Additionally, both slides have the same instructional role – they provide examples of some specific features of XML Schema. Let us assume that a student requested a learning content on XML Schema and the system has started executing algorithm presented in Section 4.1. Obviously, the slides from Fig. 6 will be in the set of the CUs retrieved from the LO repository in the first step of the algorithm. To create a coherent learning content out of the collected CUs, the system has to determine how to properly sequence those CUs. Proper sequencing assumes: 1) sequential introduction of complexity – simple concepts should always be introduced before complex topics, 2) respect of the student's learning style, particularly, in the context of our example, some students prefer to be first presented with definitions and then provided with examples of a domain



Example: Declaring XML Schema Elements

Element Contains only Child Elements

```
<xsd:element name="Person">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="Title" type="xsd:string"/>
        <xsd:element name="FirstName" type="xsd:string"/>
        <xsd:element name="Surname" type="xsd:string"/>
      </xsd:sequence>
    </xsd:complexType>
</xsd:element>
```

a)

Example: Declaring XML Schema Elements

Element Contains Simple Content and Attributes

```
<xsd:element name="apple">
  <xsd:complexType>
    <xsd:simpleContent>
      <xsd:extension base="xsd:string">
        <xsd:attribute name="variety"
                  type="xsd:string"
                  use="required"/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
</xsd:element>
```

b)

Fig. 6 Sample slides annotated with the XML Schema domain ontology concept

topic, whereas others are inclined towards the opposite approach. Semantic annotations of CUs are the primary source of information for resolving the problem of proper sequencing. In particular, the most relevant are: *dc:subject* metadata element pointing to a concept from the domain ontology and *alocom-meta:type* element pointing to the formal representation of the instructional role of a CU (i.e. concept from the ALOCoMCT ontology). Since the domain ontology only has 'XML Schema' concept to represent any content related to this very broad topic, it is clear that both slides from Fig. 6 will have the same value for the *dc:subject* metadata. Additionally, both slides have the same instructional role (*alocomct:Example*). In such a situation, the dynamic assembly subsystem can only guess the right order of the CUs. On the other hand, for people familiar with XML Schema concepts it is easy to deduce that slide (b) should precede slide (a), as comprehension of the example from slide (b) is a prerequisite for understanding the example on slide (a). However, the system does not know this, as its sole source of knowledge is the IIS domain ontology that does not contain detailed knowledge about the XML Schema concept.

To resolve this problem we need a more precise formal description of the IIS domain. In other words, the employed domain ontology needs to be significantly enlarged: each leaf class of the current ontology should be substituted with a set of concepts and relationships that describe the domain topic more precisely. Accordingly, we intend to organize the domain ontology in modules, including the core part (the IIS domain ontology in its current state) and a number of extensions, one for each complex concept of the current ontology. The OWL ontology language, we used to encode the IIS ontology,provides support for such a modular approach. Additionally, each extension of the domain ontology needs to be accompanied by a corresponding extension of the LP ontology defining an optimal learning path through the concepts of the extension. Finally, TANGRAM's subsystem for automatic semantic annotation of CUs needs to be improved if we want to fully exploit the potentials that semantically rich domain ontology offers. Although the initial evaluations of this subsystem proved to be rather satisfactory, our intention is to further improve it with more advanced text mining and information extraction techniques.

6 Related Work

Farell et al. have developed the Dynamic Assembly Engine (DAE), aimed at automatic assembly of LOs into simple, short, focused, Web-based custom courses [5]. The process is based upon the learner's request and consists of searching a LOR for relevant LOs and sequencing the retrieved LOs into a coherent learning path. Being partially inspired by the work of Farrell et al., our approach to dynamic content assembly exhibits some common traits with theirs'. Nonetheless, as TANGRAM is based on a content structure ontology (ALOCoMCS ontology), it enables reuse of CUs of different granularity levels. In other words, TANGRAM allows one to reuse not only LOs (as DAE does), but also smaller CUs (COs and CFs). Furthermore, unlike our system, DAE does not keep the users data relevant for content adaptation (e.g. learning style, preferences, knowledge of the domain topics). Instead the adaptation is based exclusively on the user's request, i.e. keyword query, desired level of detail, and the amount of time available for learning. Like TANGRAM, DAE uses its own profile of the IEEE LOM metadata schema. However, while TANGRAM's profile is used to annotate both LOs and their components (i.e. reusable CUs of divers granularity levels), in DAE the developed profile is used exclusively for annotating LOs. Another similarity of the two systems lies in their usage of a domain ontology for semantic annotation of LOs. Furthermore, the two systems use similar taxonomies to annotate LOs with instructional roles.

OntAWare provides an environment comprising a set of software tools that support learning content authoring, management and delivery [8]. It enables semi-automatic generation of LOs out of appropriate domain ontologies. Actually, LOs are produced by the application of graph transformations to these ontologies. However, since ontologies are aimed primarily for machine (not human) consumption, they typically contain terse and often scarce, human-readable descriptions of concepts and their relationships. Therefore, content generated solely from a domain ontology can be used as a skeleton for a LO, rather than as a LO per se. Further, adaptation of learning content is of a limited scope and is based solely on a student's browsing history – a track of domain concepts presented to the student during his/her single session with the system. Students' personal traits are not considered at all. Additionally, the algorithm for dynamic composition of LOs is hard-coded, making it difficult to change the instructional approach to content authoring. Learning Paths ontology makes such a change in TANGRAM much easier.

Henze [7] has developed a framework for creating and maintaining Personal Readers that provide personalized contextual information on the currently considered LO, like recommendations about additional readings, more general/detailed information, exercises, quizzes, etc. The driving principle of this framework is to expose different personalization functionalities as services which are coordinated by a mediator service. Each personalization service performs a specific kind of a LO personalization, based on the LO's metadata, user's characteristics and an appropriate domain ontology. At the current state, Personal Reader employs a very simple user model that keeps track of the learning resources the user has visited. LO's metadata must be fully IEEE LOM compliant, if it is to be processed by the system. Concepts of the domain ontology are used to enhance LOs annotations with semantic metadata. The flexibility offered by such a service-oriented architecture, made us rethink the current design of our system and made it service oriented.

7 Conclusion

The paper presents an approach to dynamic assembly of personalized learning content using the Semantic Web technologies. The peculiarity of our approach is that we reuse existing content units of different granularity levels to dynamically generate new learning content compliant to the specific needs of each individual student. To evaluate the feasibility of the proposed approach we developed TANGRAM, a web-based learning environment for the domain of Intelligent Information Systems. TANGRAM enables on-the-fly assembly of new learning content compliant to the student's knowledge of the subject domain, his/her preferences and learning style. Furthermore, TANGRAM allows quick access to a particular type of content about a domain topic of interest. Although TANGRAM supports exclusively the domain of IIS, it can be easily repurposed for other domains if appropriate domain ontology and its related learning path ontology are provided.

While working on TANGRAM's implementation we became aware of same important practical details concerning dynamic assembly of CUs originating from different sources (i.e. LOs) - for example, the problem of ordering of CUs dealing with the same domain concept. In our future research we address this issue by defining a richer domain ontology, as well as by further improving TANGRAM's subsystem for automatic semantic annotation of CUs. We also plan to extend our solution to enable repurposing content of other types of LOs beside slide presentations.

References

1. Brooks, C., McCalla, G., and Winter, M., "Flexible Learning Object Metadata", In Proc. of the Int'l Workshop on Applications of Semantic Web Technologies for E-Learning, Amsterdam, The Netherlands, 2005.
2. Brusilovsky, P, "Methods and Techniques of Adaptive Hypermedia," Adaptive Hypertext and Hypermedia, Kluwer Academic Publishers, the Netherlands, 1998, pp. 1-43.
3. De Bra, P., Aroyo, L., and Cristea, A., Adaptive Web-based Educational Hypermedia, Book chapter in: Web Dynamics, Adaptive to Change in Content, Size, Topology and Use, (Eds.) Mark Levene, Alexandra Poulovassilis, pp. 387-410, Springer, 2004
4. Dolog, P., and Nejdl, W., "Challenges and Benefits of the Semantic Web for User Modeling," In Proc. of AH2003 Workshop at 12th Int'l WWW Conf., Budapest, Hungary, May 2003.
5. Farrell, R., Liburd, S. D., and Thomas, J. C., "Dynamic Assembly of Learning Objects," In Proc. of the 13th Int'l WWW Conf., New York, USA, 2004, pp. 162-169.
6. Felder, R., and Silverman, L., "Learning and Teaching Styles In Engineering Education," Journal of Engineering Education, Vol.78, No.7, pp. 674–681, 1988.
7. Henze, N., "Personal Readers: Personalized Learning Object Readers for the Semantic Web," *Proc. of the 12th Int'l Conf. on Artificial Intelligence in Education*, Amsterdam, The Netherlands, 2005.

8. Holohan, E., Melia, M., McMullen, D., and Pahl, C., “Adaptive E-Learning Content Generation based on Semantic Web Technology”, In Proc. Int'l Workshop on Applications of Semantic Web Technologies for E-Learning, Amsterdam, The Netherlands, 2005
9. Jovanović, J., Gašević, D., and Devedžić, V., “TANGRAM: An Ontology-based Learning Environment for Intelligent Information Systems,” Proc. of the 10th World ELearn Conf, Vancouver, Canada, 2005, pp. 2966-2971.
10. Jovanović, J., Gašević, D., Verbert, K., and Duval, E., “Ontology of learning object content structure,” *Proc. of the 12th Int’l Conf. on Artificial Intelligence in Education*, Amsterdam, The Netherlands, 2005, pp.322-329.
11. Keenoy, K., Levene, M., & Peterson, D., (2003.) “Personalisation and Trails in Self e-Learning Networks”, SeLeNe Working Package 4 Deliverable 4.2. [Online]. Available at: http://www.dcs.bbk.ac.uk/selene/reports/Del4.2-2.1.pdf.
12. Robert, J.-M., and Gingras, G., "Experimental study on the reuse of learning objects and teaching practices", Proc. Int’l Conf. on Education and Technology, Calgary, Canada, 2005.
13. Stojanović, Lj. et al. (2001). “eLearning in the Semantic Web,” Proc. of the WWW2001 Int’l Conf., Orlando, USA.
14. Verbert, K., Klerkx, J., Meire, M., Najjar, J., and Duval, E., “Towards a Global Component Architecture for Learning Objects: an Ontology Based Approach,” *Proc. of OTM 2004 Workshop on Ontologies, Semantics and E-learning,* Agia Napa, Cyprus, 2004.