

# TANGRAM for Personalized Learning Using the Semantic Web Technologies

Jelena Jovanović

University of Belgrade, Belgrade, Serbia

Email: jeljov@gmail.com

Dragan Gašević

Athabasca University, Canada

Email: dgasevic@acm.org

Vladan Devedžić

University of Belgrade, Belgrade, Serbia

Email: devedzic@etf.rs

**Abstract**— Motivated with the goal to provide dynamic assembly and personalization of learning content parts, we propose an ontology-based solution implemented as an integrated learning environment called TANGRAM. TANGRAM relies on two ontologies for representing learning object (LO) content structure and LO content type (i.e. pedagogical role). LO content described by those two ontologies is further annotated with concepts of a domain ontology, while a learning paths ontology is used to specify pedagogical relations (e.g. prerequisites) among domain concepts. A user model ontology is defined to represent relevant information about TANGRAM's users. The paper presents the employed ontologies, in the context of user modeling and personalization. Furthermore, it describes the algorithm we defined to dynamically assemble content units into learning content personalized to the user's domain knowledge, preferences, and learning styles. We also discuss our experiences with dynamic content generation and summarize results of the conducted evaluation study. Although TANGRAM is a general-purpose learning environment, in this paper, we analyze it in the domain of intelligent information systems.

**Index Terms**—ontologies, personalized learning, semantic annotation, dynamic content assembly

## I. INTRODUCTION

Learning object (LO) reusability is a nice idea, but ask yourself the following question – How many times have you reused a LO entirely as is? For example, if you are a learner, do you typically go to the Web or a LO repository, find a LO you might be interested in, and use it as is - from its beginning to its end? Alternatively, if you are a teacher or an author of educational content, do you just find a LO prepared by someone else and include it in your teaching materials as is? From our own experience, as well as from the experience of a number of other learners and teachers we talked to, such scenarios are highly unlikely.

On the other hand, how many times have you reused parts of a LO? As a learner, how many times did you search just for a specific question/answer, or a specific

problem/solution, or just an introduction to a topic, or a formula and the like, in a more elaborated LO? As a teacher or as a content author, how many times have you reused and rearranged in your slides a diagram, or a figure, or a few bullet points, or even a couple of slides prepared for another presentation, either by you yourself or by somebody else? To our knowledge, these scenarios occur much more often and bring about the idea of reusable content units at a granularity finer than a LO as a whole [1]. However, current practices assume manual reuse of parts of a LO which is often tedious and time-consuming. It takes a lot of copy-and-paste actions, rearrangements, and readjustments when authoring a new LO.

Since current e-learning standards and specification do not provide adequate support for reusing LOs in their entirety [2][3], one cannot expect the required support for content reuse on finer granularity levels. Furthermore, these standards and specifications do not capture enough information required for advanced levels of learning process personalization, such as dynamic personalization in accordance with the students' preferences, learning styles, and/or objectives [4]. For example, even though the IEEE LOM metadata standard defines over 80 different metadata elements to be used when annotating LOs, only a couple of them are (to some extent) relevant for the personalization purposes. Furthermore, regarding the possible values of these metadata elements, the standard is, on the one hand very restrictive as it defines a confined set of allowable values, and on the other hand rather loose as these values are simple strings lacking explicit semantics.

Being aware of the abovementioned shortcomings of the present e-learning standards and specifications, we opted for an alternative approach to LOs reuse and learning process personalization. Specifically, we have developed an ontology-based approach for automatic decomposition of LOs into reusable fragments, and dynamic reassembly of such fragments into personalized learning content. To test the feasibility of the proposed approach,

we have developed TANGRAM, an intelligent learning environment for the domain of Intelligent Information Systems (IIS). 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 paper is structured as follows: in the next section, we present the main idea of the proposed ontology-based approach for personalized, on-the-fly content assembly; in sections 3 and 4, we present TANGRAM as a proof of the concept implementation of the proposed approach; section 5 provides more details about the ontologies our approach is based upon, whereas section 6 gives a detailed insight into the algorithm for creating personalized learning content out of reusable content units; in section 7 we discuss practical implementation details and experiences with dynamic generation of personalized learning content; section 8 presents the most important observations from the user evaluation study; after a summary of the related work, we conclude the paper with directions for future research.

## II. THE IDEA

*Tangram* is an ancient Chinese moving piece puzzle, consisting of seven geometric shapes that can be assembled in different ways to create more elaborated shapes (Fig. 1a). This ancient game is a suitable metaphor for the approach we implemented in TANGRAM – building new learning content out of existing components and shaping up that content differently to satisfy specific needs of individual learners. When referring to the components of existing LOs we use the term *Content Unit* (CU) (Fig. 1b).

For example, assume that a learner is browsing a LO repository, looking for some introductory material on XML. She<sup>1</sup> may find a number of LOs (e.g. HTML pages, text documents, slide presentations) covering the topic of XML. However, keyword-based and metadata-based search may return LOs that only partially match the learner's needs. For instance, introductory material may be suitably and thoroughly covered not only in those LOs annotated as introductory-level ones. So, how about semantically and automatically extracting just introductory pages, slides, examples, and so on from the documents returned by classical keyword/metadata-based search? These parts can then be assembled into one or more related new LOs (Fig. 1c), and possibly added to the collection of documents of the learner's interest for future study.

This raises several important questions:

- What makes a good candidate CU?
- Is there a common structure of a LO, so that extracting CUs from a selected LO can be automated?
- What are the criteria for assembling extracted CUs into a new LO?

- What is the granularity of CUs to be considered for reuse?
- How to harmonize in the newly generated LO the content, multiple authorship of assembled CUs, the learner's peculiarities, and instructional design issues?
- After all, where do the original LOs come from and where do the new ones go to?

Being fully aware of the multitude of answers to the above questions, as well as of the enormous complexity of a system that would attempt to cover all of these issues, we have decided to implement TANGRAM pragmatically. First of all, we wanted it to be useful to our students and us in the first place – hence, we originally built it for a course that we are teaching (our course on Intelligent information systems). This experience helped us identify the problems that we were not initially aware of, and accordingly plan modifications and extensions.

We also defined the structure of LOs, i.e. their CUs and granularity pragmatically. More specifically, we started from the fact that much of the learning material (LOs) that university teachers (including us) offer to their students takes the form of slide presentations. Hence, we focused on slide presentations as LOs.

We used Semantic Web technologies, ontologies in particular, to describe formally the structure of LOs and bring in some formal semantics into their annotations. In particular, the starting point in our approach was the classification of ontologies in the domain of e-learning [5] which differentiates 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.

Accordingly, in our approach, a LO is represented in a structural ontology compliant format, whereas concepts of a domain ontology are used to describe semantically the LO's content. In addition, the concepts from a context ontology are used to semantically mark up LOs with their pedagogical/instructional roles. The proposed approach also assumes annotation of each component of a LO, thus making individual components searchable and 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 pick up components from different LOs and dynamically, on-the-fly assemble them into a new, personalized learning content. In particular, new LOs are generated dynamically as HTML pages, starting from a LO repository that stores formally structured and semantically annotated slide presentations prepared by teachers and content authors.

To handle personalization issues, we relied on learner modeling and instruction modeling (i.e. instructional design). A user model ontology is used to enable formal representation of users' data and exchange of these data

<sup>1</sup> Since gender-neutral language tends to be imprecise and/or cumbersome, throughout the paper we arbitrarily decided to use feminine pronoun for each generic reference to a student.

with other learning applications. We have also defined a special ontology that enables a teacher to formally represent an optimal learning path through domain topics (i.e., an instructional map) that a student should follow.

### III. TANGRAM FOR STUDENTS

TANGRAM is implemented as a Web application and intended to be useful to both content authors and students interested in the domain of IIS. Two basic functionalities of the system from the students' perspective are:

- Provision of learning content adapted to a student's current level of knowledge of the domain concept of interest, her learning style, and other personal preferences.
- Quick access to a particular type of content about a topic of interest such as access to examples of RDF documents or definitions of the Semantic Web (both topics belong to the domain of IIS).

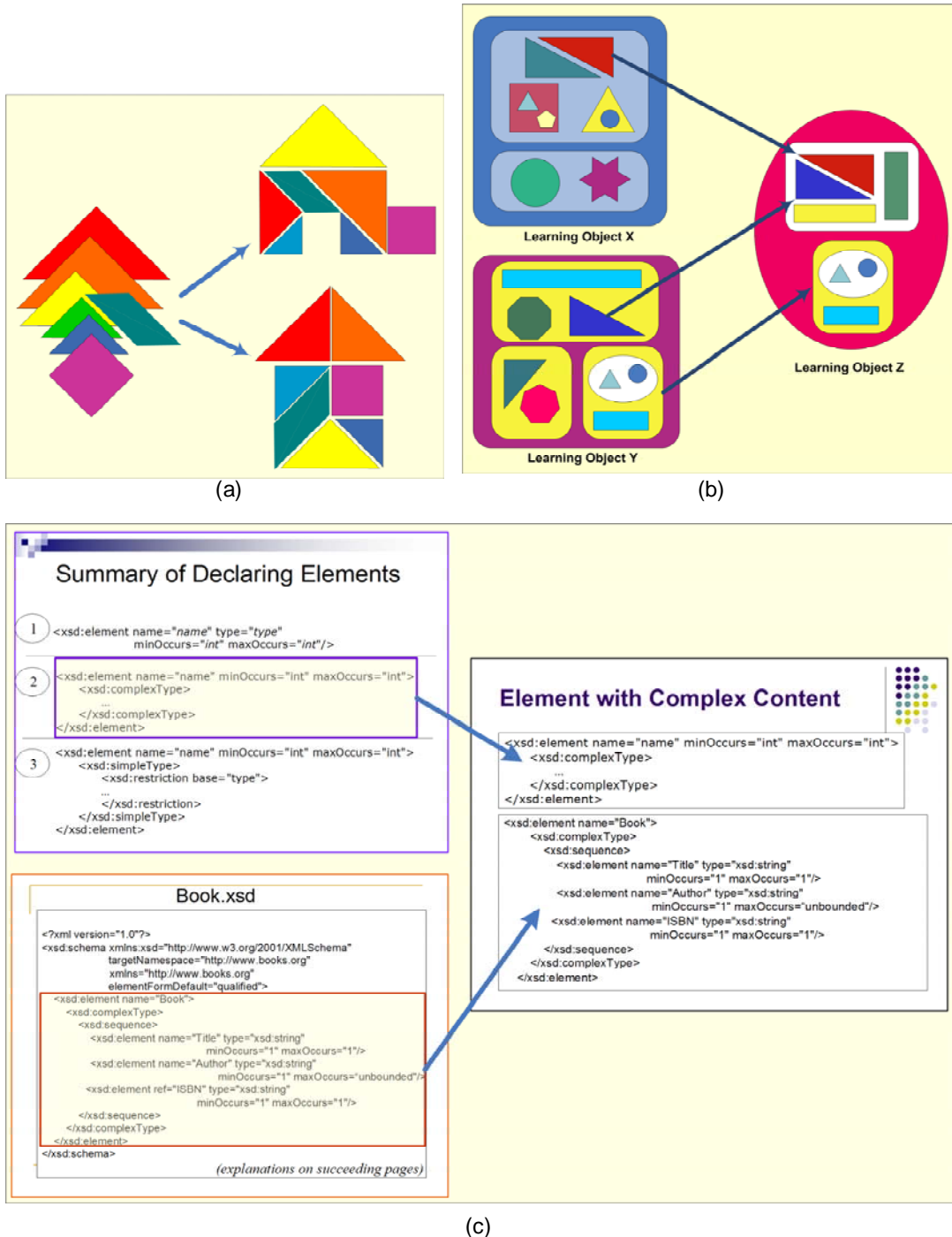


Fig.1 Tangram: a) Tangram shapes b) Reusing content units from existing learning objects c) example of learning XML – reusing parts of two slides in a third one.

In this paper, we focus on the former functionality and throughout the paper explain how it is realized in TANGRAM.

#### A. Initialization of the Student 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 her model. The student is required to fill up a questionnaire that the system uses to deduce the basic features of the student's learning style. As we had adopted the Felder & Silverman model of learning styles [6], it seemed reasonable to use the questionnaire that these authors designed for determining the learning style<sup>2</sup>. However, as the original questionnaire is rather long (44 questions), we thought it might discourage the students to use the system. Therefore, after consulting a psychologist, we made a shorter version of this questionnaire for TANGRAM. Fig. 2a is a snippet of the TANGRAM's screenshot presenting a part of the questioner.

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 her level of knowledge of the main sub-areas 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 the one that reflects her knowledge best (Fig. 2b). Internally, TANGRAM converts the student's selection for each sub-area into its numerical counterpart (0, 0.2, 0.4, 0.6 or 0.8, respectively). These numerical values are later used to let the system determine the student's initial position in the IIS domain space (more specifically, in its instructional map) and provide her with proper guidance and support.

#### B. A Learning Session

A learning session starts after a registered student selects a sub-domain of IIS to learn about, for example XML Technologies. Having verified her knowledge of the chosen sub-domain (in her user model), TANGRAM builds a visual representation of that sub-domain in the form of an annotated tree of links (the upper left corner of Fig. 3), exploiting link annotation and link hiding techniques [7]. Specifically, the following link annotations are used:

- blue bullet preceding a link to a domain concept denotes that the student knows the topic that the link points to,
- 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,
- red bullet is used to annotate a domain topic that the student is still not ready for as she is ignorant of the prerequisite topics.

The link hiding technique is used to prevent the student from accessing topics that are too advanced for her. In other words, links annotated with red bullets are made inactive. Hence, the student is free to choose one of the blue or green bulleted topics. Specifically, the annotated tree presented in Fig. 3 suggests that, according to the system's knowledge, the student has already learned about 'XML' and 'XML Schema' topics; she lacks knowledge about 'XPath', but is well prepared for learning this topic (i.e. she has all required prerequisites); the topic of 'XSLT' is still too advanced for the student as she is ignorant of the topics that are essential for successful comprehension and acquisition of knowledge on 'XSLT'. Note that TANGRAM does not aim at making a choice for the student, but at providing an adaptive guidance through the domain space and letting her decide on the topic to learn about. Let us suppose that the student wants to refresh her knowledge on the already learned topic of 'XML' (e.g., she is preparing for a test) and selects the blue bulleted link pointing to the 'XML' topic. As the selection is being made, TANGRAM starts building personalized learning assemblies out of content units available in its content repository. While doing this, the system 'bears in mind' the student's learning style, her preferences regarding content authors and her learning history. In Section VI, we explain in details the algorithm that the assembly process is based upon. After this process is finished the student is presented with brief descriptions of the built assemblies sorted according to their (calculated) relevance for the student (Fig. 3). One can notice the "Already seen" label right beneath the title of the second assembly. The system uses this label to remind the student that she studied from that assembly during some of her previous sessions with the system. Our design decision to offer a student with guidance without making a choice instead of her is obvious at this step as well. Accordingly, we provide the student with personalized selection of learning contents relevant for her present learning needs and then let her freely explore them. Suppose that the student, being aware of the fact that she has already learned from the second assembly, decides to explore the content of the first assembly now. The system presents the content of the selected assembly and updates the student model. If the student is not satisfied with the presented content or wants to explore additional contents on the same topic, she can always return to the page listing the available assemblies (Fig. 3) and selects another assembly. From this page the student can follow the link 'Back to topic selection' (in the bottom part of Fig. 3) to return to the page offering selection of sub-domains and then select some other area of the IIS domain to learn about (e.g., Intelligent Agents, Semantic Web, etc.) More details on the whole process are provided later in the paper.

#### IV. TANGRAM'S ARCHITECTURE

Fig. 4a illustrates TANGRAM's architecture. As the figure suggests, the following four main modules in TANGRAM's modular architecture are coordinated by the Coordinator module:

<sup>2</sup> The questionnaire is known as "Index of Learning Styles", and is available at <http://www.engr.ncsu.edu/learningstyles/ilsweb.html>

- *Content Management Module* is generally responsible for handling uploaded LOs and manipulating the TANGRAM's repository of LOs. The main functionalities of this module include: a) Decomposition of an uploaded LO into content units of lower granularity levels, according to the content structure ontology; b) Automatic annotation of content units – content units generated out of the uploaded LO are automatically annotated using the TANGRAM's profile of the LOM

RDF Binding. Concepts of appropriate ontologies (domain ontology and the ontology of pedagogical context) set as values of certain metadata elements bring in semantics in the content mark-up; c) Storage of LOs in a format compliant to the applied content structure ontology; d) Search of the repository and retrieval of content units of different types and levels of granularity.

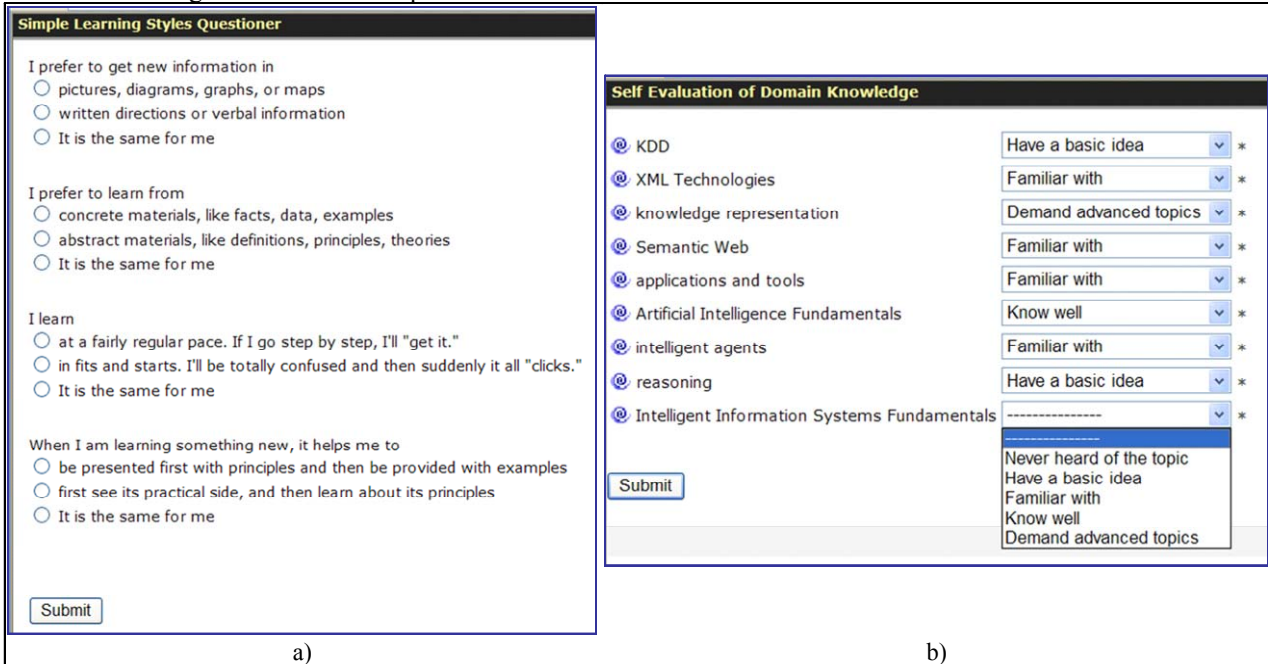


Fig. 2 A snippet of TANGRAM's screenshot presenting a part of the questioner for determining students learning style (a); a snippet of TANGRAM's page for students self-estimation of the domain knowledge.

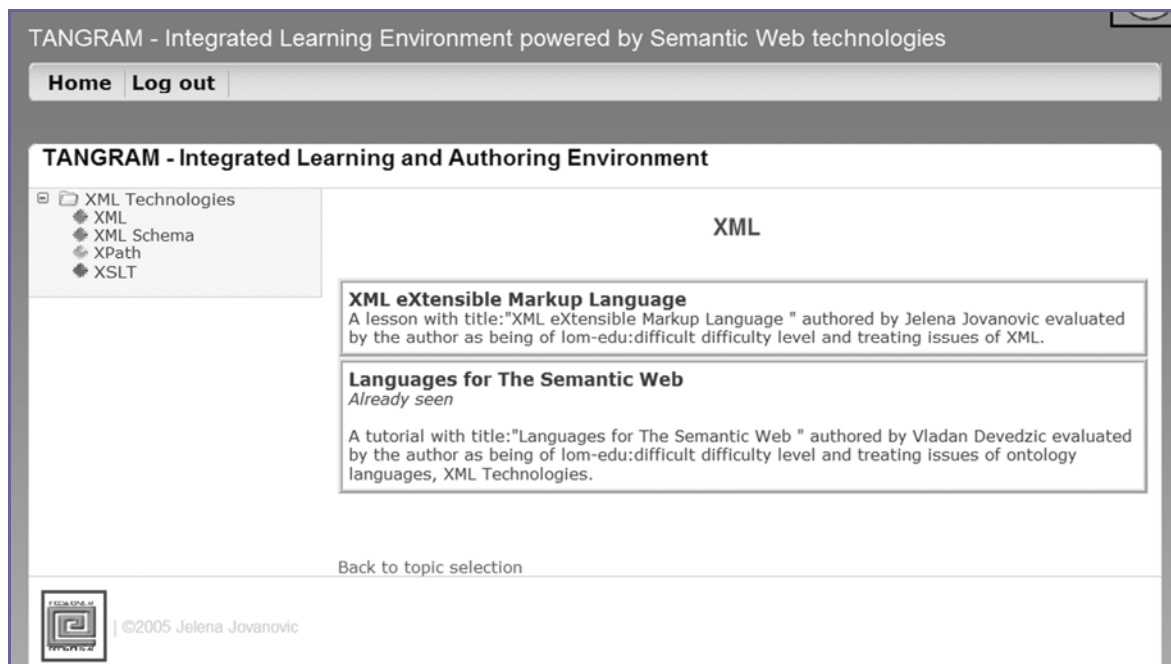


Fig. 3 A screenshot of TANGRAM after a learning topic has been chosen

- *User Model (UM) Management Module* is responsible for handling any kind of request for accessing and/or updating the repository of user models.
- *Dynamic Assembly Module* is in charge of dynamic (on-the-fly) generation of personalized learning content for a specific user (i.e. student). This module

knows how to combine available content units (obtained from the Content Management Module) to form a coherent learning content that suits a particular student best (i.e. information that the system has about the student, acquired from the UM Management Module).

- *User Interface Module* handles interaction between the system and a user.

The architecture also comprises two repositories: 1) a repository of LOs (stored in a format compliant to the content-structure ontology) and their metadata (based on the TANGRAM LOM profile); 2) a repository of user

profiles represented in accordance with the TANGRAM's User Model ontology.

The current version of TANGRAM focuses exclusively on the content structure, decomposition and annotation of slide presentations (OpenOffice and MS PowerPoint). Our decision to firstly focus on this type of LOs was motivated by the widespread use of slide presentations for organizing lecture materials and tutorials. However, our intention is to use the acquired experiences to enable decomposition and annotation of other types of LOs as well (e.g. MS Word, HTML).

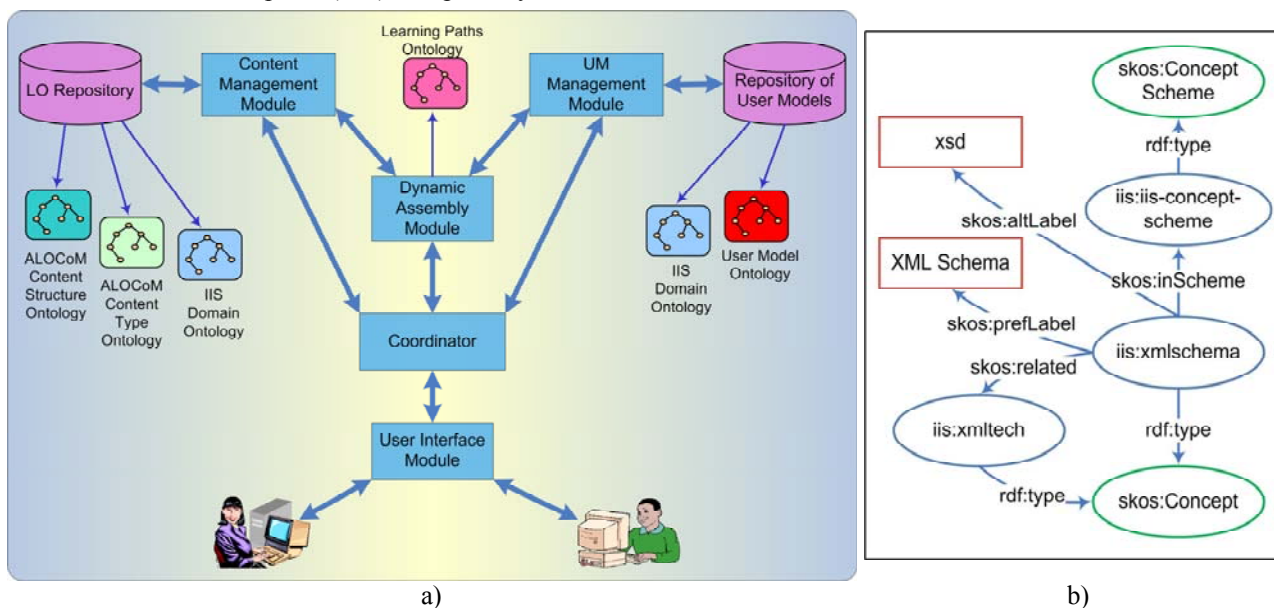


Fig. 4 TANGRAM's architecture (a); excerpt from the SKOS-based IIS domain ontology (b).

## V. ONTOLOGIES FOR DYNAMIC ASSEMBLY OF PERSONALIZED CONTENT

TANGRAM is a fully ontology-based learning environment. In the following subsections, we briefly present each of the ontologies it is based upon. These ontologies are publically available<sup>3</sup>. Additionally, to annotate content units in TANGRAM, we defined an RDF-based profile of IEEE LOM. The profile defines a subset of the IEEE LOM elements that we found necessary to support the intended functionalities of the system [8].

### A. Domain Ontology

IIS domain ontology formally defines the topics covered in our course on Intelligent information systems (such as XML technologies, intelligent agents, Semantic Web, and so on). In the development of this ontology, we used the W3C SKOS Core<sup>4</sup> ontology aimed at describing taxonomies and classification schemes. We found that SKOS Core ontology contains an excellent variety of classes (e.g. ConceptScheme, Concept) and properties (e.g. prefLabel, altLabel) to describe topics of a course. In addition, SKOS semantic properties, enabled us to struc-

ture the IIS domain in a generalization hierarchy (via the broader and its inverse narrower properties), as well as to define semantic relations between topics belonging to different branches of the hierarchy (via the related property).

Fig. 4b illustrates a part of the IIS domain ontology related to the topic of 'XML Schema'. Even without going into details of the SKOS Core ontology, this RDF graph is easy to interpret intuitively. For example, the concept *iis:xmlschema* denotes the XML Schema concept (note the link *skos:prefLabel* from *iis:xmlschema* to the "XML Schema" value-box).

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

### B. Learning Paths Ontology

The *Learning Paths (LP) ontology* specifies some aspects of learning design in TANGRAM. It defines learning trajectories through the topics defined in the TANGRAM's IIS domain ontology. The ontology is defined by extending the SKOS Core ontology with properties for defining prerequisite relationships between the domain topics (*lp:requiresKnowledgeOf* and *lp:isPrerequisiteFor*), as well as for defining the difficulty level of domain topics, as perceived by an instructor

<sup>3</sup> <http://iis.fon.rs/TANGRAM/ontologies.html>

<sup>4</sup> <http://www.w3.org/TR/skos-reference/>

(*lp:hasKnowledgePonder*). During a learning session, the learner's knowledge level is compared to the difficulty values assigned to the domain concepts by the instructor. This comparison enables TANGRAM to provide the learner with the instruction adapted to her level of mastery of domain topics.

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

In the example shown in Fig. 5, the topic *iis:xmlschema*, "XML Schema", is a prerequisite for the topic *iis:xslt*, "eXtensible Stylesheet Language (XSLT)" (the *lp:isPrerequisiteFor* link between the two topics), and the instructor has assigned to it the difficulty level of 0.4 (the *lp:hasKnowledgePonder* link between the topic *iis:xmlschema* and the corresponding value-box).

Note that the pedagogical knowledge represented by the learning paths and difficulty levels is fully decoupled from (although related to) the domain knowledge. Even if the applied pedagogical approach changes, the domain ontology remains intact and can be reused with another learning design.

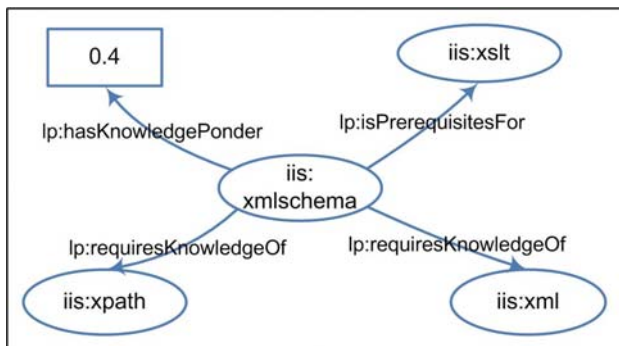


Fig. 5 An excerpt from the LP ontology

### C. Content Structure Ontology

TANGRAM's *Content Management Module* (see Section IV) decomposes the LOs that TANGRAM handles (slide presentations) into content units of various granularity levels using the *ALOCoM Content Structure ontology* (*ALOCoM CS*).

The *ALOCoM CS* ontology is based on the *Abstract Learning Object Content Model* (*ALOCoM*) [9]. Its basic concepts include *Content Fragment* (*CF*), *Content Object* (*CO*), and *Learning Object* (*LO*). *CFs* are content units in their most basic form, like text, audio, and video (i.e., raw digital resources that cannot be further decomposed into meaningful content units). A *CO* is an aggregation of *CFs* and/or other *COs*, whereas a *LO* aggregates *COs* around a single learning objective. The ontology defines classes

that formalize these basic concepts of the *ALOCoM* model, namely: *alocomcs:ContentFragment*, *alocomcs:ContentObject* and *alocomcs:LearningObject*. It also defines a number of important concepts related to the structure of almost any LO (i.e., components common to all types of LOs). These concepts are included in the ontology as subclasses of the three aforementioned root classes. In addition, we have created an extension of the core ontology which covers slide presentations' specific concepts, such as *slide* and *slide body*.

The ontology also includes properties for representing content aggregation and navigational relationships between content units. Aggregation relationships are represented in the form of *alocomcs:hasPart* and its inverse *alocomcs:isPartOf* properties. Navigational relationships between content units are specified through the *alocomcs:precedes* property and its inverse *alocomcs:follows* property.

### D. Content Type Ontology

In addition to the *ALOCoM CS ontology*, there is another important ontology in TANGRAM related to the learning content – the ontology formalizing the educational context of content units, called *ALOCoM Content Type ontology* (*ALOCoM CT*). It specifies potential instructional/pedagogical roles of content units of varying granularity levels, e.g., *abstract*, *introduction*, *definition*, *exercise*, *reference*, and so forth.

As its name suggests, this ontology is also based on the *ALOCoM* model and its basic concepts: *CFs*, *COs* and *LOs*. We identified three different kinds of instructional roles *COs* might have: cognitive (e.g. Fact, Definition, Procedure), rhetorical (e.g. Introduction, Conclusion) and supporting (e.g. Description, Example, Exercise) – these are modeled as subclasses of the *alocomcs:ContentObject* class. Concepts such as Tutorial, Lesson, and Test are introduced as subclasses of the *alocomst:LearningObject* class. The *alocomcs:ContentFragment* class is not subclassed, as according to the *ALOCoM* model [9], an instructional role cannot be assigned to a single *CF*. The creation of this ontology was mostly inspired by the research presented in [10][11]. Concepts defined in this ontology are used to annotate content units with their instructional role.

### E. 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 Learner6 and IMS LIP7. Furthermore, since we did not want to end up with yet another specific interpretation of the official specifications, potentially incompatible with existing learning applications, we explored existing solu-

<sup>5</sup> <http://dublincore.org/documents/dcmi-terms/>

<sup>6</sup> <http://edutool.com/papi>

<sup>7</sup> <http://www.imsglobal.org/profiles>

tions, like the ones presented in [12][13][14]. In addition, we established linkages with well-known Web vocabularies such as the Dublin Core and FOAF8 (Friend-Of-A-Friend). The result is a modular UM ontology that:

- uses some parts of the UM ontology developed for the ELENA project and described in [14]; 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. 6<sup>9</sup>, one can notice the class *User* that formally describes the concept of a TANGRAM user. It is defined as a subclass of the *Agent* class from the FOAF ontology. Each user can be a member of one or more organizations (*Organization*). Specifically, the user can be a member of a university (*University*) and/or a research centre (*ResearchCentre*). In addition, a user can be a member of one or more groups, such as research or study group (represented with *ResearchGroup* and *StudyGroup* classes respectively). Additionally, for each user the system needs data about her role/position in the formal organization she belongs to. Therefore, we introduced the property *hasRole* that relates an instance of the *User* class with an appropriate instance of the *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: *Teacher*, *TeachingAssistant*, *Researcher*, *Student*. Of course, this enumeration can be extended to encompass additional roles if needed. Further, each user can have certain preferences (*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 [12] and is fully compliant with the IMS LIP specification (hence *ims* prefix). The class *ims:Preference*, formally representing a user's preference, can have the *ims:hasImportanceOver* property that defines a priority of a preference (i.e. its rank in terms of importance) for a specific user. Furthermore, TANGRAM's UM ontology introduces the *AuthorPreference* class as a subclass of *ims:Preference* in order to represent users' preferences regarding authors of learning content. The property *refersToAuthor* associates this specific type of a user's preference with 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 (*Stu-*

*dent*) is assigned a set of performance-related data (via *hasPerformance* property) represented in the form of the *papi:Performance* class and the following set of properties<sup>10</sup>:

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 content unit 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 content unit 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 the 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 her learning style. The learning style of a student is formally represented by the *LearningStyle* class in the UM ontology. This class is associated (via the *hasCategory* property) with the class *LearningStyleCategory* that formally stands for one specific aspect (category) of the learning style. Specifically, TANGRAM implements the learning categories defined in 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. A subclass of the *LearningStyleCategory* class was introduced to represent each of these categories (e.g. *LS\_Visual-Verbal*). To make the ontology more general and easily extensible, we assigned the property *basedOnTheory* to the *LearningStyleCategory* class, thus enabling the introduction of learning style categories defined by other authors. The class *LearningStyleCategory* is also attached the *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 *hasValue* property of the *LS\_Visual-Verbal* class means that the learner is highly visual. On the opposite, *hasValue* property with the value of 1 identifies a highly verbal learner.

<sup>8</sup> <http://xmlns.com/foaf/0.1>

<sup>9</sup> Classes and properties that do not have namespace prefix in Fig. 6 belong to the namespace of the TANGRAM User Model ontology ([um:http://tangram/user-model/complete.owl](http://tangram/user-model/complete.owl)).

<sup>10</sup> The prefix *papi:* is used to denote that the *Performance* class and its properties are defined according to the PAPI Learner Specification.

VI. DYNAMIC ASSEMBLY OF PERSONALIZED LEARNING CONTENT

A learning session starts after a 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 her at that moment. The adaptive guidance is realized in the form of annotated tree of topic links, as the one presented in the upper left corner of Fig. 3.

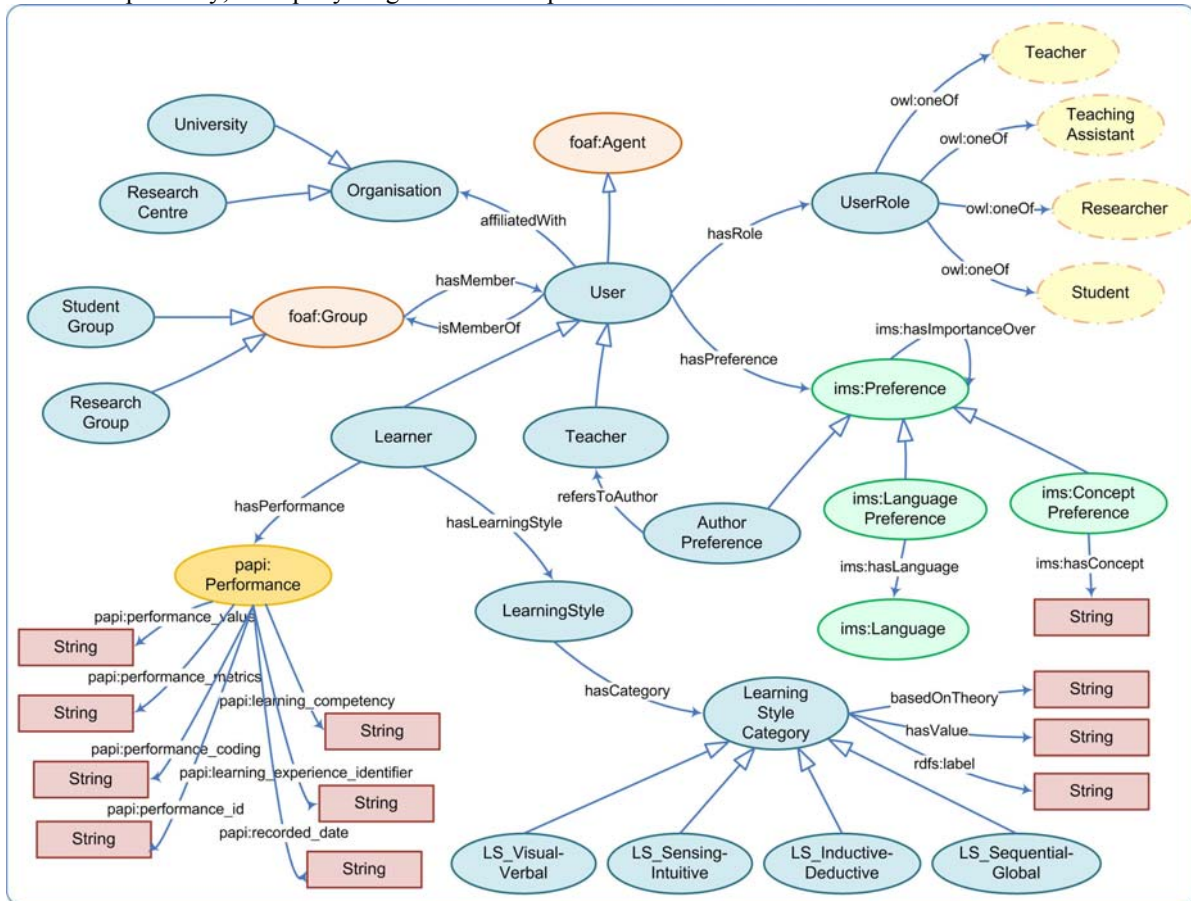


Fig. 6 Graphical representation of the TANGRAM’s User Model Ontology

After the student selects one topic 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 algorithm presented in the form of a flow chart diagram shown in Fig. 7.

The query sent to the LO Repository, in the first step of the algorithm, is based on the metadata that the content units from the repository are annotated with. Specifically, in this initial step, we retrieve the content units having the selected domain topic as the value of their `dc:subject` metadata element. If the repository does not contain content units on the selected topic, the further steps of the algorithm depend on the student’s learning style (i.e., on the system’s perception of his/her style). In particular, the system checks whether the student prefers global or sequential learning approach, i.e. it queries the student’s model for the value of the Sequential-Global learning style category (represented as an instance of the `um:LS_Sequential-Global` class on Fig. 6). Since global

learners prefer holistic approach and learn best when provided with a broader context of the topic of interest, they would even prefer to be presented with learning content dealing with advanced topics. Therefore, if the student belongs to the category of global learners, content units covering the advanced topics are retrieved from the repository and the algorithm proceeds normally. However, if the student is prone to the sequential approach, she would be confused/disoriented if the topics are not presented in a linear fashion. Hence, in the case of a sequential learner the system generates an informative message that the learning content on the selected topic is currently not available and suggests other suitable topics.

The sorting procedure, in the third step of the algorithm, is based on the original order of content units from the group, i.e. on the value of the `alocomcs:precedes` and `alocomcs:follows` properties (see Section V.C) of the considered content units. In the subsequent text, we use the

term *assembly* to refer to a group of content units sorted in this manner.

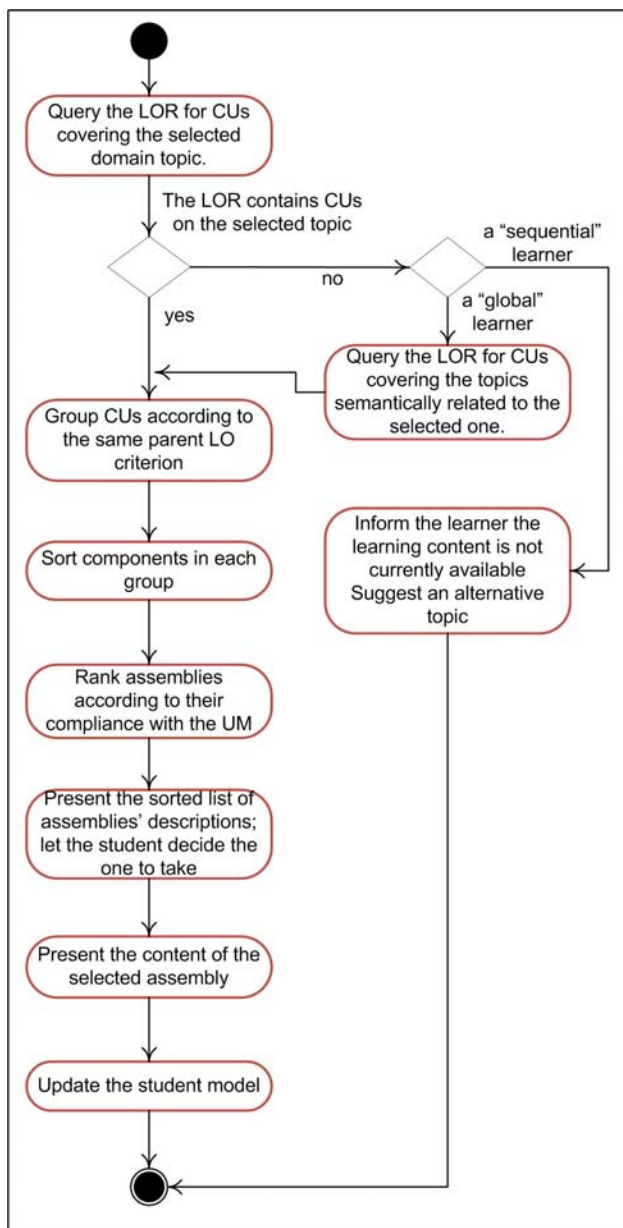


Fig. 7 The algorithm for dynamic assembly of learning content on the selected domain topic

The last but not the least important step of the algorithm consists of updating the student model. Specifically, the system creates an instance of the *papi:Performance* class in the student model and assigns values to its properties in the following manner:

1. Date and time when the student was presented with the selected assembly is assigned to the *papi:recorded\_date* property;
2. For each content unit of the assembly a *papi:learning\_experience\_identifier* property is created with the content unit's URI as its value; and
3. Domain concept(s) covered by the assembly is (are) set as the value of the *papi:learning\_competency* property.

4. The *papi:performance\_value* property is assigned a value that reflects the student's level of mastery of the learning 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. Assignment of a lower value 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. This approach was inspired by the work of De Bra et al [15].

In the fourth step of the presented algorithm, assemblies of content units are ranked according to their 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, her preferred author, as well as her learning history data (i.e., already seen content units).

*Learning style.* An assembly fully compliant with the student's learning style is assigned the relevancy value of 0.8 points. Each identified divergence results in a lower value of the relevancy factor. Specifically, compliance with the learning style categories (LSC) is considered in the following manner:

- Visual-Verbal LSC: whether the assembly is better suited for a visual or for a verbal student is determined by comparing the number of the assembly's components comprising visual elements (images) with the total number of its components. If the 'visual' components make up more than 50% of the whole assembly, the assembly is assumed to be more suitable for visual than for verbal students. If the identified visual-verbal feature of the assembly is compliant with this aspect of the student's learning style, the relevancy is increased by 0.2.
- Sensing-Intuitive LSC. Information about this feature of the assembly is obtained from the metadata of the assembly's parent LO (i.e. the LO that the assembly's components originates from). In particular, the value of the *lom-cls:accessibilityRestrictions* metadata element<sup>11</sup> is compared with the equivalent data from the student model, that is, with the value assigned to the *um:hasValue* property of the *um:LS\_Sensing-Intuitive* class. If the two values are equal or close enough (currently we consider the range of  $\pm 0.15$  as 'close'), the relevancy value is increased by 0.2.
- Inductive-Deductive LSC. The same as for the Sensing-Intuitive LSC.
- Sequential-Global LSC. If the assembly deals with advanced/related topics, it is assumed to be more suitable for global learners. As with the other LSCs, if the matching with the student's profile was found, 0.2 points are added to the assembly's relevancy score.

We did not consider Active-Reflective LSC, as it emphasizes social aspects of a learning process: an active

<sup>11</sup> We use this metadata element to specify some features of the student's learning style that the LO is suitable for. The range of this element in our profile is restricted to the instances of the *um:LearningStyle* class defined in the TANGRAM's UM ontology

student tends to retain and understand information best by doing something active with it (discussing/applying it or explaining it to others), whereas a reflective learner prefers to think about it quietly, working on her own [6]. Since TANGRAM currently does not provide support for collaborative learning, this LSC is left out of scope for the moment.

*Preference regarding the content author.* If the author of the assembly (i.e., the content units included in the assembly) is one of the student's preferred authors, the relevancy score of the assembly is increased by 0.2 points. Otherwise, the repository of user models is queried for associations between the author of the assembly and the student's preferred author(s). If some sort of close partnership can be mined (team-mates, i.e. members of the same *um:ResearchGroup*), the relevancy factor of the assembly is increased by 0.1.

*Learning history data.* Actually, this set of data about the content units that the student has already seen does not influence the value of the relevancy factor. Nonetheless, these data are used to enrich the description of the assembly. If the assembly contains completely identical set of components as an assembly that has already been presented to the student in one of her previous sessions with the system, the assembly is annotated as already seen (i.e., the 'Already seen' statement is added to its description).

In the subsequent 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.

## VII. DISCUSSION

Current implementation of the algorithm explained in the previous section uses exclusively slides (instances of the *alocomcs:Slide* class) for dynamic generation of personalized learning content. Our attempts to base the assembly process on content units of lower granularity levels (e.g., *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 content units from diverse LOs. Nonetheless, this objective turned out as overly ambitious: fully automatic 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 content unit's content as the major obstacle for using small-size content units in the process of automatic content assembly. To make these statements clearer, let us consider a small example. Fig. 8 presents two slides from different slide presentations, authored by different

authors, but covering the same domain concept – the concept of 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 the previous section. Obviously, the slides from Fig. 8 will be in the set of the content units retrieved from the LO repository in the first step of the algorithm. To create a coherent learning content out of the collected content units, the system has to determine how to properly sequence them. Proper sequencing assumes: 1) sequential introduction of complexity – simple concepts are 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 topic, whereas others are inclined towards the opposite approach. Semantic annotations [16] of content units 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 content unit (i.e., to a concept from the ALO-CoM CT ontology). Since the domain ontology only has the 'XML Schema' concept to represent any content related to this very broad topic, it is clear that both presented slides will have the same value for the *dc:subject* metadata. Additionally, both slides have the same instructional role (i.e. *alocomct:Example*). In such a situation, the dynamic assembly subsystem can only guess the right order of the content units. On the other hand, for learners familiar with XML Schema 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 (see Section V) 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 extended: 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<sup>12</sup> ontology language which 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 instructional map (compliant with the LP ontology) defining an optimal learning path through the concepts of the extension. Finally, TANGRAM's subsystem for automatic semantic annotation of content units needs to be improved if we

<sup>12</sup> <http://www.w3.org/TR/owl-ref/>

want to fully exploit the potentials that semantically rich domain ontology offers. Although the initial evaluation of this subsystem proved to be rather satisfactory [8], our intention is to further improve it with more advanced text

mining and information extraction techniques. We also intend to make use of the students' collaborative tags as an important source of semantic markup.

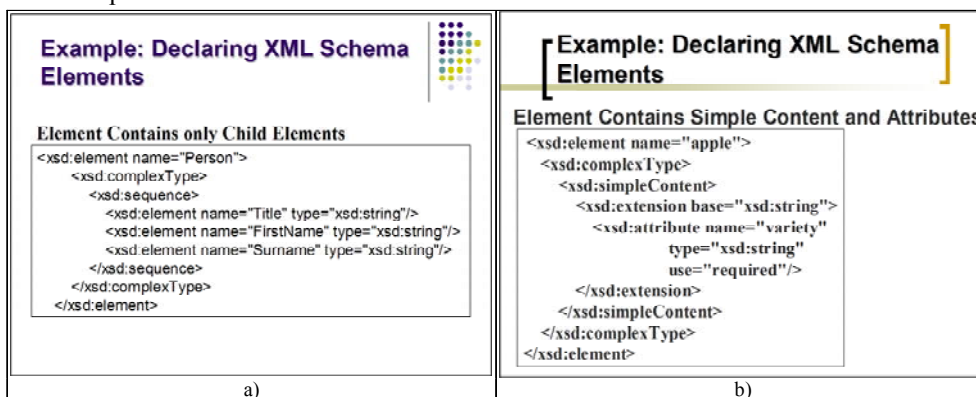


Fig. 8. Sample slides annotated with the XML Schema domain ontology concept

In the last couple of years, the Semantic Web research community has made a significant effort to disambiguate and formalize tags, that is, to bridge the gap between the needed level of semantic richness and the level offered by tags. For example, the work presented in [17] offers an interesting and comprehensive approach to semi-automatic generation of ontologies out of folksonomies. Besides being beneficial for improving semantic richness of tags, the proposed techniques could also be applied for analyzing tags to identify students sub-communities based on shared interests: annotated LOs and/or tags used for annotation. The Meaning Of A Tag (MOAT<sup>13</sup>) project is a representative of a different approach: instead of trying to disambiguate and semantically enrich tags after their creation, MOAT aims to empower users to define meaning(s) of their tag(s) – by relating them to the URIs of existing concepts from Semantic Web knowledge bases (such as DBpedia<sup>14</sup> and GeoNames<sup>15</sup>) – while they are annotating web resources [18]. While users can still benefit from the simplicity of free-tagging when annotating content, the linking to existing concepts (i.e. their URIs) offers a way to solve tagging ambiguity. Moreover, the relationships between concepts that tags are linked to can be leveraged for deducing additional relationships among tags themselves, as well as among tagged resources.

Another important direction of the improvement is to make use of the experience of students in the interaction with the suggested content units. Being inspired by the ecological approach to e-learning [19], we are considering developing a feature for learners to initially select different proposed content units in a composition of a content assembly. Based on the accumulated experience of many students, we will be able to recommend those content units (slides), which will have previously been selected by other learners with similar personal traits (e.g., learning styles and background).

TANGRAM's subsystem that performs dynamic assembly of personalized learning content is designed to be fully domain independent and only partially dependent on the instructional/pedagogical approach. Domain independence is achieved by using appropriate domain ontology as the system's only source of knowledge about the subject domain. Specifically, the current implementation of TANGRAM is based on the ontology covering the IIS domain. However, the system can be equally well applied in any other domain provided that appropriate SKOS-based domain ontology is made available. Partial independence of the instructional approach is attained by using instructional map structured in accordance with the learning paths ontology to inform the system about optimal learning trajectories through the domain concepts. However, content adaptation according to a student's learning styles and preferences is hard-coded, and thus not easily modifiable. We are currently trying to improve this aspect of the system and make it more flexible. To that end we are considering application of the FOSP method suggested by Kravcik in [20]. This method is aimed at development of adaptation strategies independent of the subject domain and instructional design.

VIII. EVALUATION

TANGRAM is in use in the Department of Software Engineering at the University of Belgrade for more than two years now. We have conducted a students' evaluation of TANGRAM and have received mainly positive reactions. The number of students enrolled in our course on Intelligent information systems is usually around 30, which statistically does not make a high reliability of the results of the evaluation. However, it was highly valuable to us as it pointed out both the strengths and, what is even more important, the weaknesses of the system. We have also constantly tracked the learners' interactions with the system through log files. In what follows we present the most important observations and the conclusions that we reached so far.

By analyzing the TANGRAM's log files that we collected in the spring semester of 2008 (with 27 students

<sup>13</sup> <http://moat-project.org>

<sup>14</sup> <http://dbpedia.org/>

<sup>15</sup> <http://www.geonames.org/>

taking the course), we clearly distinguished two groups of students. The first one includes those students who primarily use TANGRAM to learn from dynamically built course materials. The second group includes the students who tend to almost exclusively use TANGRAM to quickly access specific types of content (i.e., examples, definitions, and references) on the selected domain topic. A subsequently conducted questionnaire helped us explain this observation. It turned out that the later group was formed of students who regularly attended classes and hence used TANGRAM only as a handy way of refreshing their memory on the course's topics. The former group gathered students who, not being able to attend the classes, used TANGRAM as the primary learning facility. However, regardless of the main purpose they used TANGRAM for, students from both groups found the tool handy and useful.

Still, some students reported that from time to time they were frustrated for not being able to get access to some advanced topics of the sub-domain they had chosen to learn about. Likewise, others reported of occasionally being provided with too many pictures but not enough accompanying text (i.e., verbal explanations). Initially, we thought the problem lied in the method we use to recognize the students' learning style (see Section III.A). In other words, we had expected that the students' preferred learning styles would be more-or-less propagated throughout the course, regardless of the course topics. However, after some further analysis (comparing the collected log files and the students' responses on the questionnaire), it turned out that the students' individual interests in particular sub-domains governed their learning needs. This finding directed us to search for an alternative way of determining the students' learning needs.

The idea of breaking LOs in pieces and reusing the pieces to automatically compose new, personalized LOs was not only useful to our students, but also seems to be very attractive to a number of our colleagues (teachers) to whom we have demonstrated TANGRAM. Many of them expressed their interest in using its shell to develop support for their own courses. However, it was necessary to provide an important additional explanation to a majority of them – reusing the TANGRAM approach in another course requires the corresponding domain and the learning paths ontologies to be developed first. This involves the work of an ontology engineer, a domain expert, and an instructional designer. Since at the Department the teachers are at the same time content authors, domain experts and instructional designers, they all have the knowledge required to use TANGRAM. However, they still need external help from someone who would know how to formally express that knowledge, i.e. build domain and learning paths ontologies. In addition, the two parties must also work together in subsequent modifications/extensions of these ontologies. As with all ontology-based systems, approaches and tools that automate ontology development and annotation would be more than useful, as they would considerably facilitate further course developments based on TANGRAM.

In this context, we have recently conducted an empirical study involving 27 university level educators in Canada. In the study, we evaluated the present level of the ontology development tools (specifically, ontology learning based tools) for the development of domain ontologies for different courses. Analysis of the data collected in the study [21] will set our future research in the development of ontology engineering environments for technology-enhanced learning.

## IX. 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 [11]. The process is based upon the learner's request and consists of searching a LO repository 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 (ALOCoM CS ontology), it attempts at reuse of content units of different granularity levels. In other words, TANGRAM allows one to reuse not only LOs (as DAE does), but also smaller content units (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 for content annotation. However, while TANGRAM's profile is used to annotate both LOs and their components (i.e. reusable content units 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 their instructional roles.

*OntAWare* provides an environment comprising a set of software tools that support learning content authoring, management and delivery [22]. 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 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 et al [23][24] 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. 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 make it service oriented.

Originating from the work of Personal Reader, the GroupMe! system combines Web 2.0 and Semantic Web technologies in providing a personalized content management in a group (social networking) context [25]. From the Web 2.0 side, it leverages intuitive user interfaces that allow users to create groups of resources (Web pages, videos, images). Creation of groups, addition of resources to the groups, and any other operation related to the groups are all saved as RDF triples compliant to a set of ontologies that GroupMe! uses. Such an approach to capturing of group-related annotations leverages Semantic Web technologies for integration and sharing of resources relevant for a group of users. In particular, this eliminates the problems of ambiguity and improves the ranking of the discovered resources. However, unlike TANGRAM, GroupMe! does not address the problem of personalization. This could be a promising research topic for the future research – allowing for grouping parts of learning objects in a social networking environment.

Following the above research direction and originating from the ALOCoM ontology, Semantic Document Management System (SDMS) is proposed for managing semantic documents [26]. This system is integrated into Microsoft Office in order for users to be able to make use of semantically-enabled services and benefit from the enhancements of the well-known and proven user interfaces for document authoring and management. Content authors via their social networking relations, represented in the FOAF ontology, can exchange their semantic documents, and similarly to TANGRAM, search for parts of semantic documents. By leveraging principles of the Social Semantic Desktop, SDSM allows for peer-to-peer exchange and retrieval of parts of documents. This could be another possible direction for the future development of TANGRAM, where each content author or learner could contribute to the repository of LOs. However, unlike TANGRAM, SDMS does not provide any advance personalization based on learner models.

In the project Teachware on Demand, Hollfelder et al [27] developed an infrastructure for automatic generation of courses out of reusable content fragments. Their main idea is very similar to that of ours: the existing course material is segmented into so called learning fragments (i.e. self-contained units of content) which are annotated with metadata and stored in a repository to be used for 'on demand' assembly of new courses. Each fragment is assigned one or more concepts from a domain taxonomy to represent its 'prerequisites' and 'learning outcomes'. The course compilation algorithm works basically on the pre-knowledge conditions of fragments: a user issues a request in the form of a query specifying concepts to be taught and restrictions (e.g., author) and for the specified concepts, any pre-knowledge concepts are retrieved and appropriate fragments are selected. Unlike TANGRAM, this system does not aim at being user adaptive. Furthermore, annotation of content fragments is fully manual. In addition, everything is encoded in pure XML, hence only syntactic interoperability is accomplished.

One of the main objectives of the EU project APOSDLE is to develop a system that would be able to provide knowledge workers with learning resources relevant for their present work context [28]. In particular, based on the immediate work context of a user, the system should identify his/her missing competencies and learning needs and suggest appropriate learning resources. These learning resources are created on-the-fly from a variety of resources (documents, videos, expert profiles, and so on) already stored in the organizational memory. As in TANGRAM, in the APOSDLE project, the focus is also on reuse of content units of low granularity levels (e.g., a paragraph, an image, a page). The approach for content preparation for reuse is also similar: existing documents are segmented into smaller parts and semantically annotated with concepts from a domain ontology and an ontology of instructional roles. The difference is that their approach fully relies on the manual segmentation of documents and annotation of their components.

## X. CONCLUSIONS

The paper presents a novel approach to learning content reuse and personalization based on the Semantic Web technologies, ontologies in particular. Specifically, we argue for reusing content units of different levels of granularity 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, her preferences and learning style. Furthermore, TANGRAM allows for 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.

The development of TANGRAM helped us reveal some important practical details of dynamic content assembly that we were not aware of initially. The most challenging task proved to be the proper sequencing of small size content units, as well as meaningful arrangement of different kinds of content, authoring styles, terminology, and other relevant features. We still have not found a proper solution to this problem, but we are currently trying to resolve it by developing a more detailed formal description (i.e., ontology) of the IIS domain. We intend to use it to more precisely semantically describe LO's components, thus enabling TANGRAM to make better informed decisions when performing on-the-fly content assembly. Accordingly, the TANGRAM's subsystem for automatic semantic annotation of content units needs to be further improved if we want to fully exploit the potentials that the enriched domain ontology will offer. We also intend to extend TANGRAM with support for collaborative tagging and use students' collaborative tags to allow the system to make more informed decisions when generating personalized learning content.

We also plan to extend our solution to enable repurposing content of other types of LOs beside slide presentations; currently we are working on the support for OpenOffice documents.

Last but not the least important, we intend to address the issue of domain ontology evolution – as a course evolves, the domain ontology for that course has to evolve too, so that the semantics it captures do not lag behind the course. In our ongoing research efforts, we are developing a novel method of interactive visualizations that provides an intuitive and practical way for instructors to use the implicit feedback available from student folksonomies for evolving domain ontologies [29]. In addition, we have developed a method which uses algorithms for computing the semantic relatedness to further facilitate the teacher's task of ontology maintenance by suggesting him/her the tags that are relevant for any particular ontology concept [30]. By combining these two methods, we aim at providing support for ontology evolution which is consistent with the course content, and with the conceptualizations that instructors and students have of that content.

#### REFERENCES

- [1] K. Verbert, and E. Duval, "ALOCOM: a generic content model for learning objects," *International Journal on Digital Libraries*, Vol. 9, No. 1, pp. 41-63, 2008.
- [2] C. Brooks, G. McCalla, and M. Winter, "Flexible Learning Object Metadata", In *Proc. of the Int'l SW-EL Workshop*, Amsterdam, The Netherlands, 2005.
- [3] N. Friesen, "Final Report on the "International LOM Survey," TR 36C087, Canadian Advisory Committee for ISO/IEC JTC1/SC36, 2004 [Online]. Available at: <http://jtc1sc36.org/doc/36N0871.pdf>
- [4] D. Gašević, J. Jovanović, and V. Devedžić, "Ontology-based Annotation of Learning Object Content," *Interactive Learning Environments*, Vol. 15, No.1, pp. 1-26, 2007.
- [5] Lj. Stojanović, S. Staab, and R. Studer, "eLearning in the Semantic Web," In *Proceedings of the International World Wide Web Conference*, Orlando, USA, 2001.
- [6] R. Felder, and L. Silverman, "Learning and Teaching Styles In Engineering Education," *Journal of Engineering Education*, Vol.78, No.7, pp. 674–681, 1988.
- [7] P. Brusilovsky, "Methods and Techniques of Adaptive Hypermedia," *Adaptive Hypertext and Hypermedia*, Kluwer Academic Publishers, the Netherlands, 1998, pp. 1-43.
- [8] J. Jovanović, D. Gašević, and V. Devedžić, "Ontology-based Automatic Annotation of Learning Content," *Int'l Journal on Semantic Web and Information Systems*, Vol.2, No.2, pp.91-119, 2006.
- [9] K. Verbert, J. Klerkx, M. Meire, J. Najjar, and E. Duval, "Towards a Global Component Architecture for Learning Objects: an Ontology Based Approach," In *Proc. of OTM 2004 Workshop on Ontologies, Semantics and E-learning*, Agia Napa, Cyprus, 2004.
- [10] C. Ullrich, "The learning-resource-type is dead, long live the learning- resource-type!," *Learning Objects and Learning Designs*, Vol.1, No.1, 2005, pp.7-15.
- [11] R. Farrell, S. D. Liburd, and J. C. Thomas, "Dynamic Assembly of Learning Objects," In *Proceedings of the 13th Int'l WWW Conference*, New York, USA, 2004, pp. 162-169.
- [12] P. Dolog, and W. Nejdl, "Challenges and Benefits of the Semantic Web for User Modeling," In *Proceedings of AH2003 Workshop at 12th Int'l WWW Conference*, Budapest, Hungary, May 2003.
- [13] K. Keenoy, M. Levene, & D. Peterson, "Personalisation and Trails in Self e-Learning Networks", *SeLeNe Working Package 4 Deliverable 4.2*. 2005. [Online]. Available at: [http://www.dcs.bbk.ac.uk/selene/reports/Del4\\_2-2\\_1.pdf](http://www.dcs.bbk.ac.uk/selene/reports/Del4_2-2_1.pdf)
- [14] P. Dolog, B. Simon, T. Klobucar, and W. Nejdl, "Personalizing access to learning networks," *ACM Transactions on Internet Technologies*, Vol. 8, No. 2, Article No. 3, 2008.
- [15] P. De Bra, L. Aroyo, and A. Cristea, "Adaptive Web-based Educational Hypermedia," in: *Web Dynamics, Adaptive to Change in Content, Size, Topology and Use*, M. Levene and A. Poulouvasilis, Eds. Springer, 2004, pp. 387-410.
- [16] D. Gašević, J. Jovanović, and V. Devedžić, "Ontology-based Annotation of Learning Object Content," *Interactive Learning Environments*, Vol. 15, No.1, pp. 1-26, 2007.
- [17] C. Van Damme, M. Hepp, and K. Siorpaes, "FolksOntology: An Integrated Approach for Turning Folksonomies into Ontologies," In *Proceedings of the Workshop Bridging the Gap between Semantic Web and Web 2.0*, 2007, pp.57-70.
- [18] A. Passant, and P. Laublet, "Meaning Of A Tag: A collaborative approach to bridge the gap between tagging and Linked Data," In *Proceedings of the WWW 2008 Workshop Linked Data on the Web (LDOW2008)*, Beijing, China, 2008.
- [19] G. McCalla, "The Ecological Approach to the Design of E-Learning Environments: Purpose-based Capture and Use of Information About Learners," *Journal of Interactive Media in Education*, 2004 (7).
- [20] M. Kravcik, "Specification of Adaptation Strategy by FOSP Method," In *the Proceedings of AH2004 Conference, Eidhoven, Netherland, 2004*.
- [21] M. Hatala, D. Gašević, M. Siadaty, J. Jovanović, and C. Torniai, "Can Educators Develop Ontologies Using Ontology Extraction Tools: an End User Study," *4<sup>th</sup> European Conference on Technology-Enhanced Learning*, Nice, France, 2009, unpublished.
- [22] E. Holohan, M. Melia, D. McMullen, and C. Pahl, "Adaptive E-Learning Content Generation based on Semantic Web Technology", In *Proceedings of Int'l SW-EL Workshop, Amsterdam, The Netherlands, 2005*.

- [23] F. Abel, I. Brunkhorst, N. Henze, D. Krause, K. Mushtaq, P. Nasirifard, and K. Tomaschewski, "Personal Reader Agent: Personalized Access to Configurable Web Services," *In Proceedings of 14th Workshop on Adaptivity and User Modeling in Interactive Systems*, Hildesheim, October 9-11 2006.
- [24] N. Henze, "Personal Readers: Personalized Learning Object Readers for the Semantic Web," *In Proceedings of the 12th Int'l Conf. on Artificial Intelligence in Education*, Amsterdam, The Netherlands, 2005.
- [25] F. Abel, N. Henze, D. Krause, and M. Kriesell, "Semantic Enhancement of Social Tagging Systems," *Annals of Information Systems*, Special Issue on "Semantic Web and Web 2.0", Springer, in press.
- [26] S. Nešić, "Semantic Document Model to Enhance Data and Knowledge Interoperability," *Annals of Information Systems*, Special Issue on "Semantic Web & Web 2.0", Springer, in press.
- [27] S. Hollfelder, S. Rodolfo, and P. Fankhauser, "Tailoring Training Courses Using XML-based Metadata," *DELOS Workshop on Interoperability in Digital Libraries, in conjunction with ECDL*, Darmstadt, September, 2001.
- [28] C. Ghidini, V. Pammer, P. Scheir, L. Serafini, and S. Lindstaedt, "APOSLE: Learn@work with semantic web technology," *in I-Know '07*, Graz, Austria, 5-7 September 2007.
- [29] C. Torniai, J. Jovanović, D. Gašević, S. Bateman, and M. Hatala, "E-learning meets the Social Semantic Web," *In Proc. of the 8th IEEE Int'l Conference on Advanced Learning Technologies (ICALT 2008)*, Santander, Cantabria, Spain, 2008, pp. 389-393.
- [30] C. Torniai, J. Jovanović, S. Bateman, D. Gašević, and M. Hatala, "Leveraging Folksonomies for Ontology Evolution in E-learning Environments," *In Proceedings of the 2nd IEEE International Conference on Semantic Computing*, Santa Clara, CA, USA, 2008, pp. 206-215.

**Jelena Jovanović** is an Assistant Professor of Computer Science with the Department of Software Engineering, FON - School of Business Organization, University of Belgrade, Belgrade, Serbia. She received her B.S., M.Sc. and PhD degrees in

informatics and software engineering from University of Belgrade in 2003, 2005 and 2007, respectively. Her research interests are in the areas of semantic technologies, Web technologies, technology-enhanced learning, and personalized learning. She is a member of the GOOD OLD AI research group. She can be reached at <http://jelenajovanovic.net>.

**Dragan Gašević** is a Canada Research Chair in Semantic Technologies and an Assistant Professor in the School of Computing and Information Systems at Athabasca University. He is also an Adjunct Professor in the School of Interactive Arts and Technology at Simon Fraser University Surrey and an associated research member of the GOOD OLD AI Research Group at the University of Belgrade. He is a recipient of Alberta Ingenuity's 2008 New Faculty Award. His research interests include semantic technologies, software language engineering, technology-enhanced learning, and service-oriented architectures. He has (co-)authored more than 200 research papers. He has been serving on editorial boards of five international journals and has edited special issues in journals such as IET Software and IEEE TSE. He has been the organizer, chair, and member of program committees of many international conferences. He can be reached at <http://dgasevic.athabascau.ca>.

**Vladan Devedžić** received the B.S., M.S. and PhD degrees in computer science from the Department of Computer Science, School of Electrical Engineering, University of Belgrade, Serbia, in 1982, 1988, and 1993, respectively. He is a professor at and the chair of the Department of Software Engineering, FON - School of Business Administration, University of Belgrade, Serbia. His main research interests include software engineering, intelligent systems, knowledge representation, ontologies, Semantic Web, intelligent reasoning, and applications of artificial intelligence techniques to education and medicine. So far, he has authored/co-authored more than 290 research papers, several book chapters and 6 books. He has been serving on editorial and reviewing boards of several international journals. He can be reached at <http://devedzic.fon.rs/>.