Ontology-Based Automatic Annotation of Learning Content

Jelena Jovanović, University of Belgrade, Serbia and Montenegro

Dragan Gašević, Simon Fraser University Surrey, Canada

Vladan Devedžić, University of Belgrade, Serbia and Montenegro

ABSTRACT

This paper presents an ontology-based approach to automatic annotation of learning objects' (LOs) content units that we tested in TANGRAM, an integrated learning environment for the domain of Intelligent Information Systems. The approach does not primarily focus on automatic annotation of entire LOs, as other relevant solutions do. Instead, it provides a solution for automatic metadata generation for LOs' components (i.e., smaller, potentially reusable, content units). Here we mainly report on the content-mining algorithms and heuristics applied for determining values of certain metadata elements used to annotate content units. Specifically, the focus is on the following elements: title, description, unique identifier, subject (based on a domain ontology), and pedagogical role (based on an ontology of pedagogical roles). Additionally, as TANGRAM is grounded on an LO content structure ontology that drives the process of an LO decomposition into its constituent content units, each thus generated content unit is implicitly semantically annotated with its role/position in the LO's structure. Employing such semantic annotations, TANGRAM allows assembling content units into new LOs personalized to the users' goals, preferences, and learning styles. In order to provide the evaluation of the proposed solution, we describe our experiences with automatic annotation of slide presentations, one of the most common LO types.

Keywords: please provide

INTRODUCTION

Over the past few years we have witnessed a tremendous amount of activity taking place in the development of Web-based e-learning systems (Mohan & Greer, 2003). A substantial percentage of those activities have been related to learning content authoring. As authoring of high quality learning materials proved to be a highly expensive task in terms of both time and money, reuse of once created learning content soon become one of the hottest research issues. Learning content represented in the form of reusable learning objects (LOs) promised to significantly reduce the time and cost of authoring high-quality learning materials, making them more affordable and readily available. The principal objective is to enable faster, cheaper, and better learning (Duval & Hodgins, 2003).

Current research efforts are almost exclusively oriented toward reusability of LOs in their entirety. Annotations of LOs with the standard-compliant metadata sets (e.g., IEEE Learning Object Metadata [LOM, 2002] [LOM] and Dublin Core) aim at enabling search and retrieval of existing LOs stored in LO repositories. Accordingly, metadata is seen as the primary mean for fostering LOs reusability. However, very often a content author needs to reuse just some specific parts of an LO, rather than the entire LO, for example, just a couple of slides out of a slide presentation, or an image or a table out of a text document. Faced with such a need, the content author typically turns to what we call the *search-read-copy-paste* approach. Specifically, the process of authoring new learning materials typically proceeds in the following steps: an author first searches both LO repositories and the Web to find potentially useful learning content. Then (s)he reads the retrieved materials to determine whether they really contain content relevant for the course under development. Having recognized relevant parts of the retrieved materials, the author copies/pastes them in the new materials (s)he is authoring. The process finishes by fine-tuning content units collected from different sources and optionally adding some new, original contents. Obviously, the content authoring process demands an LOt of time and effort. Additionally, it is not scalable in terms of maintenance (Verbert, Jovanović, Gašević, & Duval, 2005). (Semi-)automating reuse of individual components of LOs' can improve the current practice by reducing the effort that content authors put in preparation of learning materials. However, an approach to such a kind of automation is still an open question.

To enable reusability of content units of varying granularity levels, an explicit definition of the LO's structure is needed. Additionally, if the process of reusing content units has to be (semi-)automatic, the definition of the LO's structure must be formally specified and expressed in a machine understandable language. Furthermore, to facilitate search and retrieval of content units based on the semantics of their content, those content units must be semantically annotated, that is, semantic metadata must be attached to them. Ontologies and Semantic Web languages provide means to achieve both things.

In this paper we present our approach to automatic annotation of LOs' components based on a number of ontologies. The approach is tested in TANGRAM — an integrated learning environment for the domain of Intelligent Information Systems (IIS).

PROBLEM STATEMENT

The objectives of this paper are:

- To present the rational for using Semantic Web technologies, ontologies in particular, to annotate LOs and their components, and thus facilitate LOs reusability at the component (content unit) level;
- To present how automatic semantic annotation is implemented in a practical learning environment — TANGRAM — developed applying Semantic Web technologies to enable reusability at the level of LOs' components;
- To discuss our experiences with automatic annotation of individual content units.

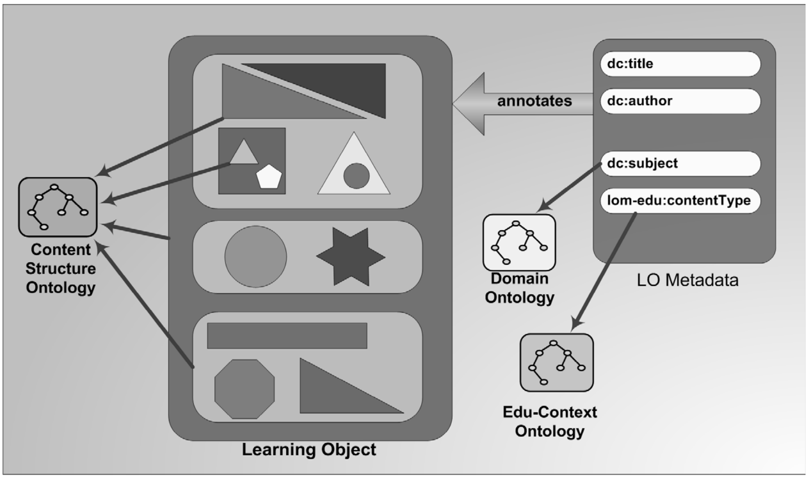
The principles we discuss are implementation-independent. On the other hand, their implementation in TANGRAM helped us reveal important practical details and problems we were not aware of initially.

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

THE RATIONAL FOR SEMANTIC ANNOTATION OF LOS

The starting point in our approach to ontology-based LO annotation is the classifi-

Figure 1. An LO compliant to the proposed ontology-based approach



cation of ontologies relevant for the e-learning domain suggested by Stojanović, Staab, and Studer (2001). This classification recognizes the following types of ontologies: (1) structural ontologies that formalize the content structure; (2) context ontologies that specify the pedagogical/instructional role of the content; (3) content (domain) ontologies that formally describe the subject matter (topics) of learning content. In our approach an LO is represented by a structural ontology, whereas the other two types of ontologies are used to semantically annotate the LO. Figure 1 illustrates the proposed approach. Each LO is considered as an aggregate consisting of a number of content units/components. The components can differ in types and levels of granularity. The concepts of a *Content Structure Ontology* formally define different kinds of content units (e.g., slide, paragraph, list), whereas the properties of such an ontology enable formal expression of aggregation relationships between content units of different granularity and/or type. Additionally, each LO is annotated with a standards-compliant metadata set. Specifically, our proposal is based on IEEE LOM standard (LOM, 2002). However, we argue for certain enhancements of the standard in order to make the metadata machine understandable. Therefore, we suggest using domain ontology concepts as values of the metadata element describing the content of an LO — for example, we assign concepts of an ontology for the IIS domain (*Domain Ontology* in Figure 1) to the *dc:subject* element of our standards-compliant metadata schema (see the section on "TANGRAM's LOM RDF Binding Profile" for more details). In addition, the concepts from a context ontology (*Edu-Context Ontology* in Figure 1) are used to mark-up LOs with their pedagogical/instructional roles (e.g., definition, illustration). The proposed approach also assumes attaching metadata to each component of an LO, thus making individual components searchable and reusable (this detail is left out from Figure 1 in order to avoid excessive cluttering).

The rational for using ontologies in the proposed approach is to enable Semantic Web reasoners to perform an advanced search of LO repositories. The advancement reflects in ability to search for a content of a certain type (as defined in a context ontology, e.g., "definition"), dealing with a certain topic (from a domain ontology, e.g., "Semantic Web") and being at a certain level of granularity (as defined in a structure ontology, e.g., "slide"). Besides the benefit of having a more convenient search mechanism that better reflects the searchers' needs, another important benefit lies in an ability to (semi-)automatically compose the retrieved content units into a new LO compliant to the specific instructional approach of a content author.

The suggested approach is also relevant in terms of learning content personalization. Explicitly defined structure of an LO facilitates adaptation of the LO, as it enables direct access to each of its components and their tailoring to the preferences, objectives, competencies, and/or other specific features of a student that are relevant for the learning process. Besides, being able to directly access components of an LO, we are empowered to dynamically, on the fly create a new, personalized learning content out of those components.

WHAT IS TANGRAM?

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. This ancient game perfectly reflects the basic notion of the approach we propose — building new content out of existing components and shaping up that content differently to satisfy specific needs of individual learners. Accordingly, we gave the name TANGRAM to the application we are developing to evaluate the feasibility of our approach. TANGRAM is a learning Web application intended to be useful to both content authors and students interested in the domain of IIS. In the rest of this section we first describe TANGRAM from two different view points, content authors' and students', and then proceed with presenting its architecture.

What does TANGRAM Provide to a Content Author?

TANGRAM's aim is to enable content authors to create new LOs out of existing learning content with as little manual operations (copy, paste) as possible. To this end, TANGRAM aims at providing the following functionalities:

- Upload a new LO into the LO Repository with the idea of later being able to reuse its components. The uploaded LO is decomposed into smaller content units in accordance with the used content structure ontology. The idea is to make each content unit directly accessible, thus facilitating its reuse.
- Describe the uploaded LO and its components with high-quality metadata, but without too much effort for the author. Annotation is based on a subset of the IEEE LOM metadata elements (LOM, 2002), actually only those elements that we found necessary to provide intended functionalities of our system.
- Search the LO Repository for LOs and/or their components in order to employ them for composing new LOs.
- Compose a new LO using components previously retrieved from the repository.

To be able to use the system, an author has to register first. We made the registration mandatory in order to acquire a basic set of data about the author. Availability of such data facilitates generation of suggested values for metadata elements in the process of LOs annotation.

What does TANGRAM Provide to a Student?

TANGRAM provides adaptation of learning content to the specific needs of individual students. Currently, TANGRAM is focused on enabling personalized learning experiences 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, for example, access to *example*s of RDF documents or *definition*s of the Semantic Web (both topics belong to the domain of IIS).

Just like a content author, a student also 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 profile (i.e., student model). The learner's learning style is determined from a simplified version of the Felder and Silverman questionnaire[1], whereas for determining the learner's initial knowledge about the IIS domain, the system relies on the learner's self-assessment. The system uses this profile to keep track of the student's preferences, learning style, as well as his/her level of knowledge about concepts from the IIS domain. With this data, the system is able to create personalized learning content (see the "Ontology-Based Approach to Personalization of Learning Content" section).

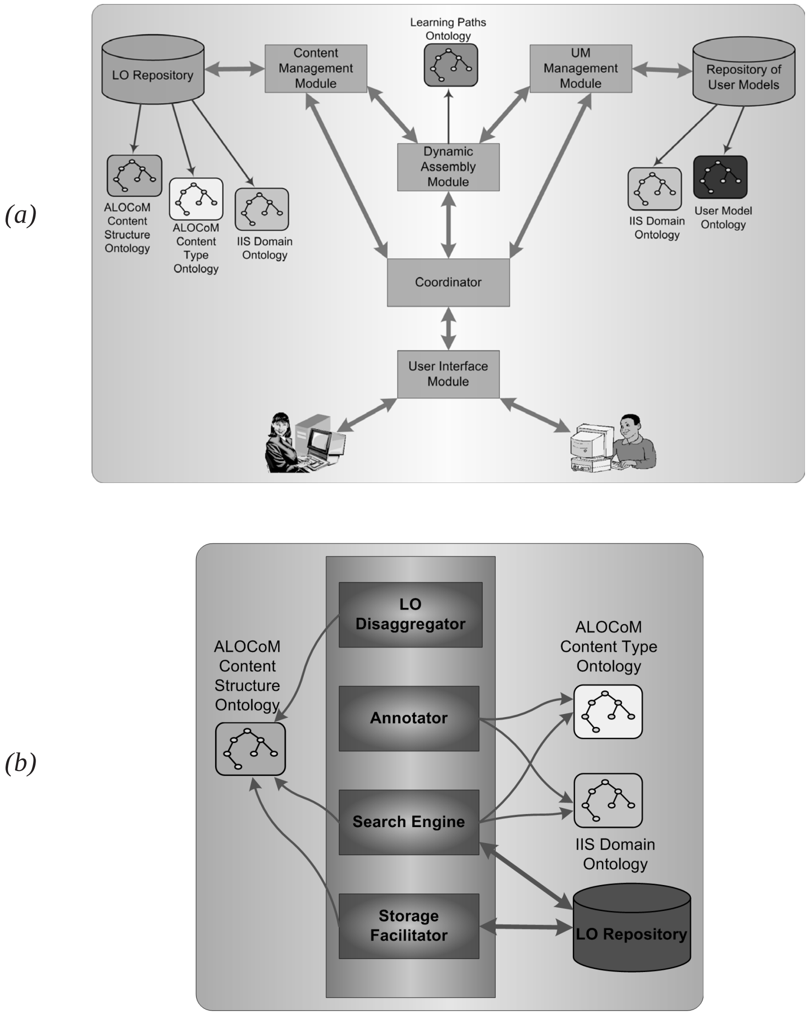
TANGRAM'S Architecture

Figure 2a illustrates TANGRAM's architecture. As the figure suggests, TANGRAM has a modular architecture, comprised of the following four main modules coordinated by the *Coordinator* module:

- *Content Management Module* is generally responsible for handling uploaded LOs and manipulating TANGRAM's repository of LOs. Figure 2b illustrates the architecture of this module, whose main functionalities include: (a) Decomposition of an uploaded LO into content units of lower granularity levels, according to the content structure ontology (*LO Disaggregator*); (b) Automatic annotation of content units (*Annotator*) — content units generated out of the uploaded LO are automatically annotated with metadata elements of TANGRAM's IEEE LOM RDF Binding profile (see the section on "TANGRAM's LOM RDF Binding Profile" for details). Concepts of appropriate ontologies (domain ontology and the ontology of pedagogical context), set as values of certain metadata elements, facilitate automatic interpretation of the semantics (i.e., meaning) of the content mark-up; (c) Storage of LOs in a format compliant to the applied content structure ontology (*Storage Facilitator*); (d) Semantic search of the repository and retrieval of content units of a specific type, and/or dealing with a specific domain topic (*Search Engine*).
- *User Model (UM) Management Module* is responsible for handling any kind of request for accessing and/or updating the repository of user models (profiles).
- *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 current version of TANGRAM focuses exclusively on the content structure, decomposition, and annotation of slide presentations. Specifically, TANGRAM is presently able to handle only slide presentations authored in OpenOffice, but we are in the midst of providing the same support for the MS PowerPoint authoring tool. Our decision to firstly focus on slide presentations was motivated by the fact that teachers frequently opt for this type of LO when preparing learning content for in-class lectures and tutorials. A plethora of LOs of this kind is already made available on the Web, providing a valuable source of learning content worth for reuse. 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).

Figure 2. TANGRAM's architecture (a) Content Management Module; (b) TANGRAM's 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 TANGRAM's IEEE LOM RDF Binding profile); (2) a repository of user profiles represented in accordance with TANGRAM's User Model ontology



TANGRAM is implemented in Java programming language. It is built using Tapestry (http://jakarta.apache.org/tapestry) — an open-source framework for creating dynamic, robust, highly scalable Web applications in Java. Additionally, Jena — Java Semantic Web Framework (http://jena.sourceforge.net/) — is used for storing, updating, and searching repositories of ontological instances, as well as for reasoning over the ontologies.

ONTOLOGICAL FOUNDATION OF TANGRAM

TANGRAM is a fully ontology-based learning environment. In the following subsections we briefly present each of the ontologies upon which it is based. We also describe how the ontologies are used to support personalization of learning content in TANGRAM. All ontologies are expressed in Ontology Web Language (OWL) — W3C's official recommendation for the standard ontology language. They are available at http://iis.fon.bg.ac.yu/TANGRAM/ontologies.html.

ALOCoM Content Structure Ontology

The ALOCoM Content Structure (ALOCoM CS) ontology is an extension of the Abstract Learning Object Content Model (ALOCoM) (Verbert, Klerkx, Meire, Najjar, & Duval, 2004) with certain concepts of the IBM's Darwin Information Typing Architecture (DITA)[2]. The ontology defines a number of concepts for different types of content units that form the structure of an LO. The first version of the ontology is elaborated in Jovanović, Gašević, Verbert, and Duval (2005). However, having further studied existing LO content models and content packaging formats (e.g., SCORM Content Aggregation Model — CAM[3], MPEG-21[4]), we made a major revision of the ontology and split it into two parts: an ontology of content structure and an ontology of educational content types.

The ALOCoM CS ontology distinguishes between content fragments (CFs), content objects (COs), and LOs. CFs, formalized as instances of the *alocomcs:ContentFragment* class, are content units in their most basic form (e.g., text, audio, and video), and cannot be further decomposed. COs, formally represented as instances of the *alocomcs:ContentObject* class, aggregate CFs and add navigation. Navigational elements enable sequencing of CFs in a CO. Besides CFs, COs can also include other COs. LOs (*alocomcs:LearningObject*) aggregate COs around a single learning objective. To enable more fine grained content structuring we analyzed the structure of widely used content formats (primarily slide presentations and textual documents) and identified a number of specific content structuring types (e.g., slide, slide body, title, table). These types are included in the ontology as subclasses of the three root concepts (i.e., CFs, COs, and LOs). Finally, the ontology defines 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 are specified as the *alocomcs:ordering* property that defines the order of components in a CO or an LO in the form of an *rdf:List*. Figure 3 is a graphical representation of the ontology's basic classes and properties.

ALOCoM Content Type Ontology

The ALOCoM Content Type (CT) ontology is also rooted in the ALOCoM model and has CF, CO, and LO as the basic, abstract content types. However, these concepts are now considered from the perspective of pedagogical/instructional roles they might have. Therefore, concepts like Definition, Example, Exercise, Reference are introduced as subclasses of the CO class, whereas concepts such as Tutorial, Lesson, Test are some of the subclasses of the LO class (Figure 4). The CF class is not sub-classed, as according to the ALOCoM model (Verbert et al., 2004); an instructional role can not be assigned to a single CF. Creation of this ontology was mostly inspired by a thorough examination of existing LO Content Models (such as SCORM [SCORM, 2004] or Learnativity [Wagner, 2002]) as well as by a closely related work presented in Ullrich (2005). Concepts defined in the ontology are used to annotate LOs and their components with the pedagogical/instructional role(s) for which they were intended. One should note that a CO can be assigned multiple pedagogical roles, each one defined from a different perspective: rhetorical, cognitive, supporting (Figure 4).

Figure 3. ALOCoM Content Structure ontology — Basic classes and properties

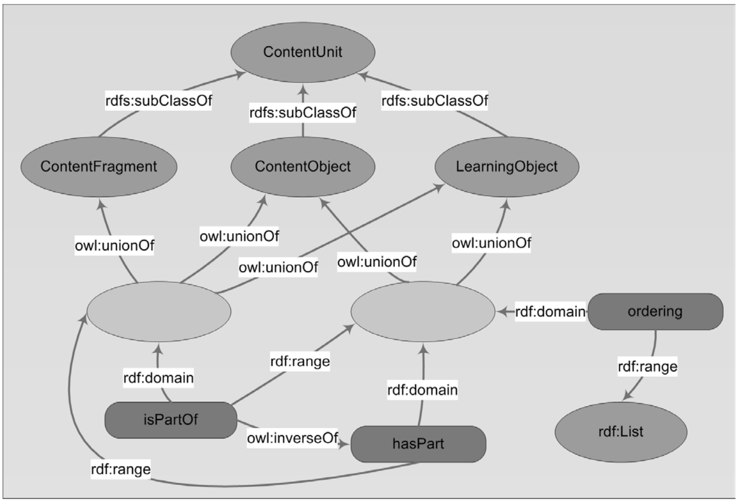
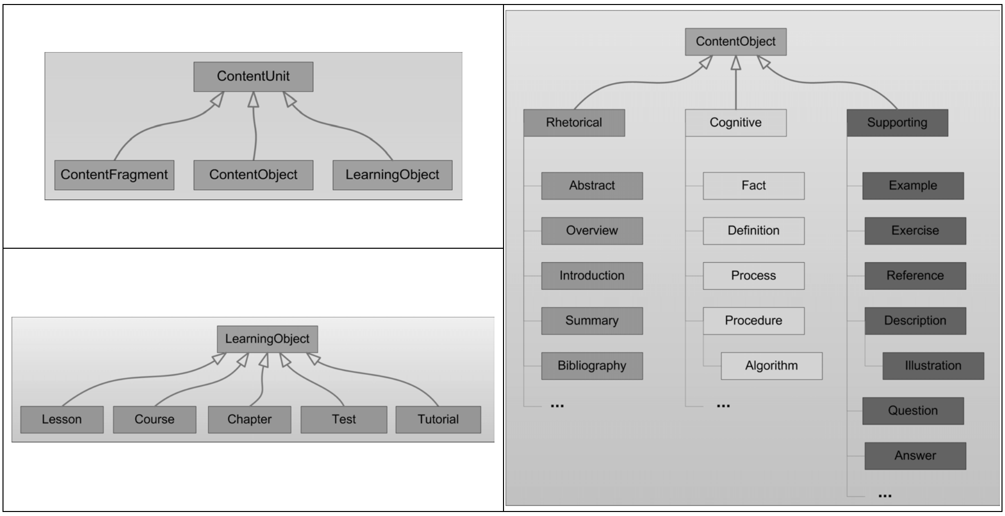


Figure 4. Graphical representation of a part of the ALOCoM CT ontology



Presently, the ALOCoM CT ontology has a rather simple structure. It is more a taxonomy than a real ontology, since it defines only a hierarchy of concepts without specifying any kind of relationships among them. Despite its simplicity, this ontology provided us with means to formally state identified pedagogical role(s) of LOs and their components. Nonetheless, our intention is to enrich the ontology with semantic properties as formal expressions of interrelations among different pedagogical roles, and hence enable an advanced level of reasoning.

Domain Ontology

The SKOS Core ontology (http://www.w3.org/2004/02/skos/core/) is used as the basis of the IIS course domain ontology. SKOS Core is a member of the SKOS family of ontologies developed through W3C's Simple Knowledge Organization System (SKOS) and the Semantic Web efforts. It is specifically developed to describe taxonomies and classification schemes and hence has an excellent variety of properties to describe relationships between topics in a course.

We used an OWL binding of the SKOS Core ontology to formally represent sub-domain of IIS[5]. Figure 5 illustrates a segment of the developed domain ontology. Each domain concept is represented as an instance of the *skos:Concept* class, while the conceptual scheme of the IIS domain is represented as an instance of the *skos:ConceptScheme* class. The *skos:inScheme* SKOS property is used to associate all defined instances of the *skos:Concept* class to the conceptual scheme of the IIS domain, that is, to the instance of the *skos:ConceptScheme* class as its formal representation. Likewise, each identified domain concept is assigned one or more aliases (terms typically used in literature when referring to a concept) using SKOS properties: *skos:prefLabel*, *skos:altLabel*, and *skos:hiddenLabel*. SKOS semantic properties, that is, properties derived from the *skos:semanticRelation* property, enabled us to structure the IIS domain in a generalization hierarchy (via *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 *skos:related* property). We used *skos:hasTopConcept* property to relate most general domain topics (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.

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 students. That kind of information is stored separately in the Learning Paths ontology.

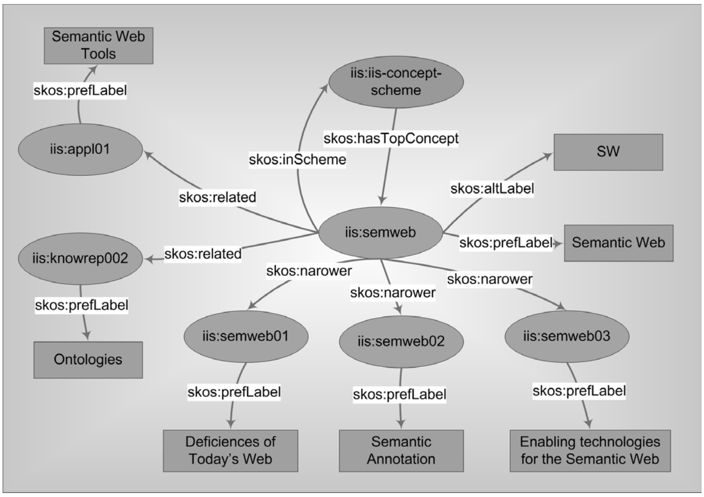
Other Developed Ontologies

Besides the above-mentioned ontologies, TANGRAM's functionalities also largely depend on the Learning Paths and the User Model ontologies. Since these two ontologies are not essential for the automatic annotation of content units, we just briefly explain them. For more details about these two ontologies and their roles in TANGRAM one can refer to Jovanović, Gašević, and Devedžić (2006).

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 LP ontology relates instances of the domain ontology through an additional set of semantic relationships reflecting a specific instructional approach to teaching/learning IIS. The main benefit of decoupling the domain knowledge from the pedagogical knowledge is to enable reuse of the domain ontology — even if the applied pedagogical approach changes, the domain ontology remains intact.

The User Model (UM) ontology formally represents relevant information about TANGRAM users (both content authors and students). To enable interoperability with other learning applications and enable exchange of users' data, we based the ontology on the official specifications for user modeling: IEEE PAPI Learner (http://edutool.com/papi/) and IMS LIP (http://www.imsglobal.org/profiles/). Specifically, the ontology focuses only on those elements of the specifications that proved to be essential for TANGRAM's functionality.

Figure 5. A segment of the domain ontology describing the concepts of the Semantic Web



Ontology-Based Approach to Personalization of Learning Content

In this section we briefly explain how TANGRAM leverages the synergy of the presented ontologies and the automatically generated semantic annotations to dynamically build personalized learning content. In other words, having presented TANGRAM's ontological foundation, we are able to provide more details on the TANGRAM's functionalities introduced in the "What Does TANGRAM Provide to a Student?" section.

A learning session starts after a user (registered and authenticated as a learner) selects a sub-domain of IIS to learn about. The system verifies the learner's knowledge of the chosen sub-domain using the data stored in the learner's model, the IIS domain ontology, and the LP ontology. Specifically, the LP ontology is queried for the prerequisite topics for the selected sub-domain (i.e., topics related via *lp:requiresKnowledgeOf* property with the sub-domain's topics). Subsequently, the learner model is queried for the learner's level of knowledge about the topics of the selected sub-domain as well as the identified set of prerequisite concepts. The acquired information enables TANGRAM to build a visual representation of the sub-domain (i.e., its hierarchical organization of concepts) in the form of an annotated tree of links, exploiting link annotation and link hiding techniques (Brusilovsky, 1998). One should note that TANGRAM does not aim to make a choice for a learner. Instead, the system provides adaptive guidance to direct the learner toward the most appropriate topics for him/her, but eventually lets him/her decide on the topic to learn and the content from which to learn.

After the learner selects a domain concept from the topics tree, on-the-fly assembly of learning content begins. Firstly, TANGRAM's repository of LOs is searched for content units covering the selected domain topic. The search is based on the *dc:subject* metadata element of the content units stored in the repository. If content units on the selected topic are not available, the learner's model is consulted for the learner's learning style, specifically for its Sequential-Global dimension. If the learner is described as a global learner, preferring holistic approach and learning best when provided with a broad context of the topic of interest (Felder & Silverman, 1988), content units covering advanced topics (as specified in the LP ontology) are used instead. Otherwise[6], the system informs the learner that the learning content on the selected topic is currently unavailable and suggests other suitable topics. Subsequently the retrieved content units are grouped according to the same parent LO criterion (following the containment hierarchy via content units' *alocomcs:isPartOf* property). Then, exploiting the *alocomcs:ordering* property of the group's parent LO, each group is sorted to reflect the original order of content units. Each sorted group (in the subsequent text referred to as *assembly*) is assigned a relevancy — a decimal number between 0 and 1 that reflects its compliance with the learner's model, that is, its relevancy for the learner. The computation of an assembly's relevancy is based on the data stored in the learner's model, such as the learner's learning style, preferences, and the learning history. Subsequently, the assemblies are sorted according to the calculated relevancy, and their descriptions are presented to the learner. 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. As the learner selects an assembly from the list, the system presents its content using its generic form for presentation of dynamically assembled learning content. Finally, the learner model is updated.

ANNOTATION OF CONTENT UNITS

The majority of metadata required for annotation of an LO are directly (manually) supplied by the content author, when uploading the LO to the repository (Figure 6). In other words, LOs are semi-automatically annotated. However, annotation of LOs' components is fully automated. In this section we firstly present the profile of the LOM RDF Binding that we developed to annotate content units in TANGRAM and then proceed to explain automatic generation of metadata for LO's components.

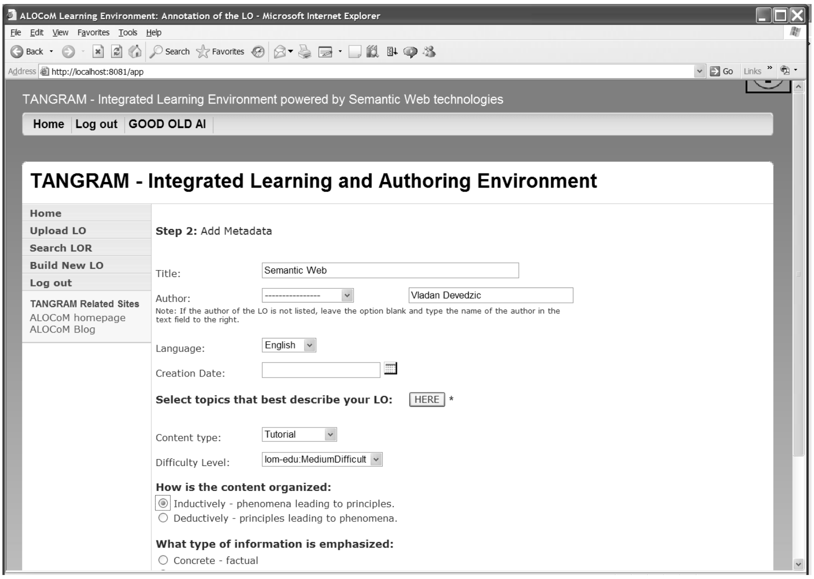
TANGRAM's LOM RDF Binding Profile

Each content unit should be annotated in order to be more easily searchable and thus reusable. Annotations of content units in TANGRAM are based on the IEEE LOM standard. However, since TANGRAM is envisioned as an application for the Semantic Web, that is, Web aimed for both human and machine consumption, all the data it deals with needed to be presented in a machine comprehensible format. This means that not only content units but also their metadata must be expressed in a Semantic Web language. Accordingly, our starting point was the official proposal for the IEEE LOM RDF Binding specification (Nilsson, 2002). However, not all LOM elements are used, but a subset necessary to support the intended functionalities of the system. In other words, we created an application profile of the LOM RDF Binding. Figure 7 illustrates elements of the profile that we made for TANGRAM.

All metadata elements presented in Figure 7 are fully compliant with the LOM RDF Binding specification, except for two elements: learning resource type and classification.

Learning Resource Type. We introduced the *alocom-meta:type* property instead of the *rdf:type* property that is suggested by the LOM RDF Binding to be used for specifying learning resource type of a content unit. The reason for this deviation from the official proposal lies in the following: we introduced the *alocom-meta:Metadata* class to represent a metadata set attached to a content unit. Since an instance of this class, representing one particular set of metadata, already has its own *rdf:type* property set (pointing to the *alocom-meta:Metadata* class, see Figure 14b), adding another property of the same type, but with different semantics (type of learning resource) would bring in a confusion. Furthermore, we do not use the LOM restricted vocabulary as values of this element, as it mixes concepts of instructional (e.g., Exercise, Simulation, Experiment) and technical (e.g., Diagram, Graph, Image) nature (Ullrich, 2005). Instead, we use concepts of the ALOCoM CT ontology, as they describe instructional aspects

Figure 6. Screenshot of TANGRAM's page for annotation of uploaded LOs


ALOCoM Learning Environment: Annotation of the LO - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost:8081/app
Go Links
TANGRAM - Integrated Learning Environment powered by Semantic Web technologies
Home Log out GOOD OLD AI
TANGRAM - Integrated Learning and Authoring Environment
Home
Upload LO
Search LOR
Build New LO
Log out
TANGRAM Related Sites
ALOCoM homepage
ALOCoM Blog

Step 2: Add Metadata

Title: Semantic Web

Author: ---------------- Vladan Devedzic

Note: If the author of the LO is not listed, leave the option blank and type the name of the author in the text field to the right.

Language: English

Creation Date:

Select topics that best describe your LO: HERE *

Content type: Tutorial

Difficulty Level: lom-edu:MediumDifficult

How is the content organized:

● Inductively - phenomena leading to principles.

○ Deductively - principles leading to phenomena.

What type of information is emphasized:

○ Concrete - factual

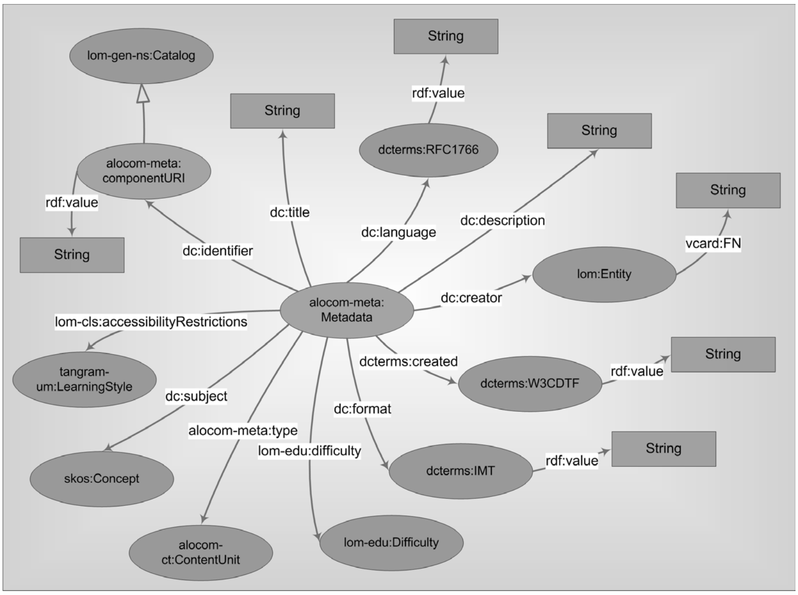
of content units, thus properly reflecting the semantics of this metadata element.

Classification. We use the *dc:subject* property to point to a concept from the domain ontology. This does not fully conform to the IEEE LOM RDF Binding since our domain ontology is not an LOM Taxonomy, as it is supposed to be according to the binding specification. Furthermore, we use *lom-cls:accessibityRestrictions* to specify some features of a student's learning style the LO is suitable for. The IEEE LOM RDF Binding specification defines this property, without imposing any specific restrictions on the domain of its use. The application of this property in TANGRAM was inspired by the work of Dolog, Gavriloaie, Nejdl, and Brase in the ELENA project (Dolog et al., 2003). They used this property to annotate an LO with access requirements expressed in terms of knowledge/competencies a student needs to have in order to access the LO. As Figure 7 shows, the range of the *lom-cls:accessibilityRestriction* property in TANGRAM's LOM RDF Binding profile is restricted to instances of the *tangram-um:LearningStyle* class defined in TANGRAM's User Model ontology.

Details of Automatic Annotation of LOs and their Components in TANGRAM

Automatic annotation of LOs' components is performed by the Content Management Module (see Figure 2) as the final step in the process of uploading a new LO to TANGRAM's repository of LOs. It is preceded by the decomposition step during which a submitted LO is disaggregated into its constituent content units. Since the ALOCoM CS ontology provides an explicit definition of the LO content structure, formally specifying both LO components and relationships between those components, it served as the foundation for the disaggregation process. Actually, this decomposition can also be regarded as a metadata generation process, since it provides content

Figure 7. TANGRAM's profile of the IEEE LOM RDF Binding



units with implicit metadata — structure related metadata.

The process of automatic annotation of LOs' components is mostly based on a *top-down* approach, meaning that metadata for describing components of an LO are derived from the metadata assigned to their parent LO. Peculiarities of this *top-down* approach can be summarized as follows:

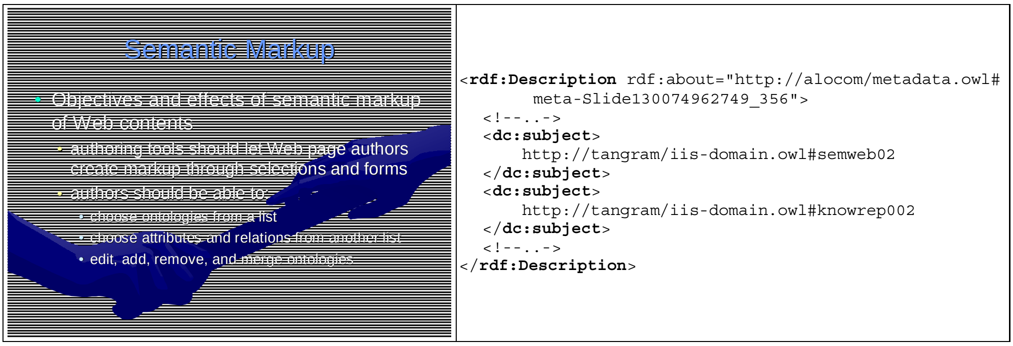
- The values of some metadata elements are literally copied from an LO to its components. This is how values are assigned to *dc:creator*, *dcterms:created*, and *dc:language* metadata elements, refereeing to the author(s), date of creation, and language(s) of a content unit, respectively.
- Some metadata elements of TANGRAM's LOM RDF Binding profile are meaningful only in the context of an LO as a whole. Therefore, they are not supposed to be assigned to the components smaller than LOs. Those metadata elements are: *lom-edu:difficulty* (difficulty level of an LO) and *lom-cls:accessibilityRestrictions* (referring to the learning styles that an LO is particularly suitable for).
- The values of the other metadata elements of a content unit are mined from its content and presentational context. In the next subsection we explain in details automatic generation of values for those metadata elements.

Mining Metadata Values

dc:title element

The *dc:title* metadata element is assigned only to COs of the *alocomcs:Slide* and *alocomcs:SlideBody* types; as for the other types of COs covered by the current version of TANGRAM, this metadata element is of no meaning. We use the text of the slide's title to assign value to the *dc:title* metadata element of the corresponding *alocomcs:Slide* and *alocomcs:SlideBody* instances. If a slide does

Figure 8. Recognition of domain concepts in a slide's content: (a) slide to be semantically marked-up; (b) a segment of the slide's metadata — Inferred values for the dc:subject element



Semantic Markup

- Objectives and effects of semantic markup of Web contents
  - authoring tools should let Web page authors create markup through selections and forms
  - authors should be able to:
    - choose ontologies from a list
    - choose attributes and relations from another list
    - edit, add, remove, and merge ontologies

```
<rdf:Description rdf:about="http://alocom/metadata.owl#
      meta-Slide130074962749_356">
   <!--..-->
   <dc:subject>
       http://tangram/iis-domain.owl#semweb02
   </dc:subject>
   <dc:subject>
       http://tangram/iis-domain.owl#knowrep002
   </dc:subject>
   <!--..-->
</rdf:Description>
```

(a) *(b)*

not have a title, these instances will not have the *dc:title* element in their metadata set.

The *dc:title* metadata element is assigned only to the one type of CFs that TANGRAM deals with, in particular, to instances of the *alocomcs:Image* class. We have noticed that authors of slide presentations very rarely use captions to describe the content (semantics) of the images appearing on their slides. In order to fill this gap, we generate a textual value that reflects the semantics of the content of an image and could serve as its caption. Therefore, if a slide (i.e., slide body, to be more precise) contains an image, we generate a "caption" for the image using the following template: "Figure <ordinal_num>. illustrating <title_of_the_slide>". The generated value is assigned to the *dc:title* metadata element of the corresponding *alocomcs:Image* instance.

Finally, it is worth mentioning that *dc:title* is one of the metadata elements of LOs that we automatically generate a value for. Its value is generated from the title of the whole slide presentation. Still, the author is given a chance to modify the generated value.

dc:subject element

To semantically annotate a CO with concept(s) from the domain ontology we apply a simple text mining approach. The starting point is the concept(s) of the domain ontology the author used to semantically markup the LO whose components (i.e., constituent COs) we intend to annotate. To illustrate this approach, let us assume that the CO to be annotated is the slide shown in Figure 8a. Additionally, we assume that this slide originates from a slide presentation (i.e., LO) manually marked-up (using the user interface shown in Figure 6) with the 'Semantic Web' concept of the Semantic Web. More technically, this means that the LO's *dc:subject* metadata element was assigned a reference to the *iis:semweb* instance of the domain ontology (presented in the center of Figure 5).

The first step is to query the domain ontology for concepts that are semantically related to the starting domain concept(s) (the concept of the Semantic Web in our example). We assumed domain concepts as semantically related if they are interconnected via the *skos:semanticRelation* property and/or its subproperties: *skos:narrower*, *skos:broader* or *skos:related* (see the "Domain Ontology" section for more details). The retrieved concepts and their aliases, that is, labels assigned to them as values of *skos:prefLabel*, *skos:altLabel* i *skos:hiddenLabel* properties, are stored in a hashmap and serve as the basis of the subsequent steps of the annotation process. Each

Figure 9. An example entry of the hashmap used in the annotation process

key: iis:semweb02
value: {Semantic Annotation, Semantic Markup, Semantic Description, Ontology-based Annotation}

entry of the created hashmap consists of a key — URI of the domain concept, and a value — a list of the concept's aliases retrieved from the domain ontology. In our example, the hashmap would contain one entry for each domain concept that is narrower in meaning than the Semantic Web concept (specifically, domain concepts with URIs: *iis:semweb01*, *iis:semweb02*, *iis:semweb03*), as well as those that are otherwise semantically related (through the *skos:related* property) to the Semantic Web (e.g., *iis:knowrep002*, *iis:appl01*) through the skos:releted property. We refer readers to Figure 5 as it illustrates the segment of the domain ontology that we discuss in this example, and thus can make the example more comprehensible. One entry of the hashmap from the example would have the form shown in Figure 9.

Subsequently each component of the slide containing text is searched for the aliases stored in the hashmap, and if some of them are found, the component (i.e., CO or CF) is annotated with the domain concepts to which the aliases refer. Afterward, we apply a *bottom-up* approach to generate a value for the slide's *dc:subject* element: the slide is annotated with a union of concepts assigned to its components. Figure 8b presents a segment of the metadata set assigned to the slide from the example (Figure 8a). As the figure shows, the slide is annotated with two domain concepts: the Semantic Annotation concept (iis:semweb02) and the Ontology concept (iis:knowrep002), since aliases of those concepts are identified in the slide's content. If no concept can be mined from the CO's content, the CO is annotated with concepts attached to the parent LO during the process of manual annotation.

For CFs that do not contain text at all, like CFs of the *alocomcs:Image* type, this approach is not applicable. Currently, in the absence of a better solution, such CFs directly inherit the value of the *dc:subject* metadata from the COs in which they are aggregated.

Furthermore, the slide presented in Figure 10a can help us explain the combined *top-down & bottom-up* approach we apply to provide values for the *dc:subject* metadata element of TANGRAM's LOM RDF Binding profile. Observing the figure, one can notice that domain concept(s) that best describe(s) the semantics of the slide's content can only be inferred from the title of the slide (the title contains one of the aliases of a domain concept). Performing the text analysis of the slide's title, we can identify XML as a domain concept that should be assigned to the *dc:subject* metadata element of the title as a content unit (i.e., to the instance of the *alocomcs:Title* class, as an ontological equivalent of the slide's title). As we have explained, applying the *bottom-up* approach we assign the same concept to the *dc:subject* metadata element of the slide that aggregates the title. Next, we analyze the content of the slide's body and each of its components, and find out that we cannot identify any concept of the domain ontology. Therefore, we apply the *top-down* approach, meaning that the XML domain concept, previously included in the slide's metadata set (via the *bottom-up* approach), is now used to semantically markup components that the slide aggregates. Specifically, in this example only the semantic annotation of the paragraph aggregated in the slide's body is really relevant, since it is the only component of the presented slide that will potentially be reused.

alocom-meta:type element

This metadata element is used to annotate LOs and COs, but not for CFs, as according to the ALOCoM model (Verbert et al., 2004) an instructional role can not be assigned to a single CF.

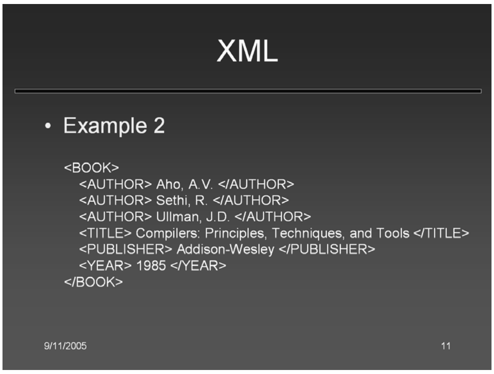
Due to the lack of well defined formats for representing learning content of a certain instructional role (e.g., an explicit format for representing definitions), we opted for a heuristics-based approach to infer instructional role of learning content units. The heuristics that we use are partially founded on our previous joint research efforts done with the ARIADNE group (http://www.ariadne-eu.org/) from K.U. Leuven, Belgium. Together, we did some initial research aimed at defining patterns for recognizing content units having instructional role of *alocomct:Definition*, *alocomct:Example*, and *alocomct:Reference* (Verbert, Jovanović, Gašević, Duval, & Meire, 2005). These patterns are defined using the experience discussed in Liu, Chin, and Ng (2003). Here, we explain how content units of type *alocomct:Example* are recognized in TANGRAM. Figure 11 presents patterns that we use to check whether a content unit is an example of a certain domain concept. In other words, these patterns enable us to test if a content unit can be marked-up with *alocomct:Example* concept as its instructional role (i.e., value of the *alocom-meta:type* metadata element).

It is important to note that the patterns shown in Figure 11 enable us to identify content units indicating the appearance of an example. In other words, they help us recognize a content unit that precedes an example. Figures 8a and 8b further explain the approach. Figure 10a shows a typical organization of a slide presenting an example of a domain concept: the title of the slide gives information about the domain concept that the example refers to, while the slide's body actually contains the example. To be more precise, the first component of the slide's body is a list (an instance of *alocomcs:List*) with only one list item (an instance of the *alocomcs:ListItem*) that, according to the pattern number 4 from Figure 11, should be classified as an example (i.e., having instructional role of an example). However, it is obvious that such a conclusion would be incorrect. Actually, the subsequent component of the slide's body — a paragraph in this case (an instance of the *alocomcs:Paragraph* concept) — should be classified as an example. On the other hand, it would be hardly possible to deduce the instructional role of this paragraph just by analyzing the text it contains. Fortunately, its structural context gives us this valuable information. In the same manner we defined and applied patterns to recognize definitions.

The slide presented in Figure 10a is suitable for explaining another specific feature of our approach to annotation of content units. When this slide is uploaded (as a part of the presentation from which it originates) to TANGRAM's repository of LOs, the Content Management Module actually stores an instance of the of the *alocomcs:Slide* class, as well as an appropriate ontological instance for each component constituting the structure of this slide (*alocomcs:Title*, *alocomcs:SlideBody*, *alocomcs:List*,...). Figure 10b provides graphical representation of the content uploaded to the repository (the slide structured according to the ALOCoM CS ontology). Furthermore, the Content Management Module uploads metadata to the repository: metadata for the slide as a whole, as well as metadata for each the slide's component that can be reused. One should note that metadata are not literally stored for every component of the slide. Instead, we store metadata only for those content units that are really reusable, in the sense that we can realistically expect someone will be interested to retrieve them form the repository and reuse. For example, in the case of the slide from Figure 10a, only metadata assigned to the slide as a whole and to the paragraph containing the text of the example will be uploaded to the repository (the ellipses highlighted in Figure 10b). The rationale is that it is highly unlikely that someone would be interested in reusing a content unit that contains only the text "Example 2" or a content unit holding the title of the slide.

Figure 10. A typical organization of a slide presenting an example of a domain concept (a); the same slide in the ALOCoM CS ontology compliant representation

(a)



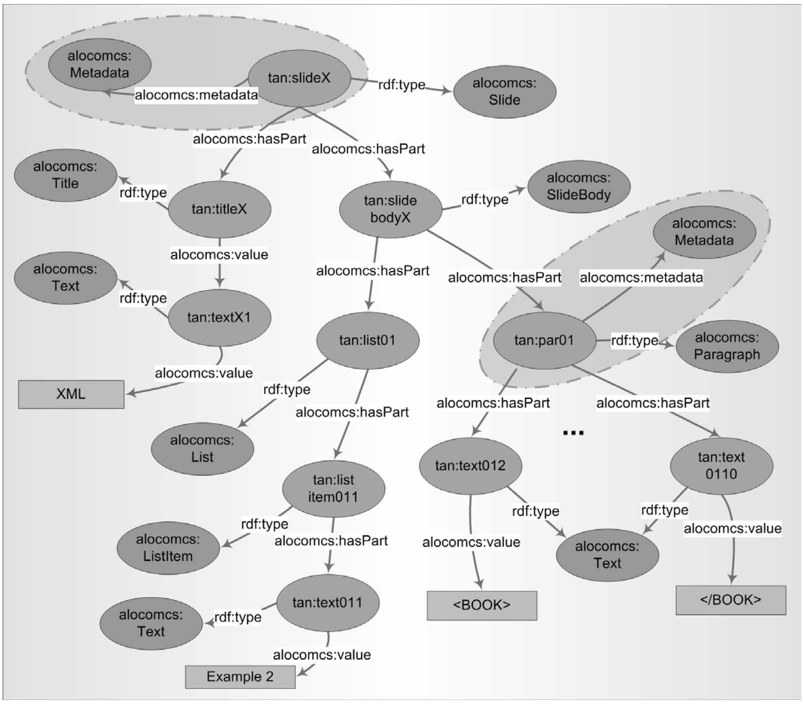
XML

- Example 2

```
<BOOK>
   <AUTHOR> Aho, A.V. </AUTHOR>
   <AUTHOR> Sethi, R. </AUTHOR>
   <AUTHOR> Ullman, J.D. </AUTHOR>
   <TITLE> Compilers: Principles, Techniques, and Tools </TITLE>
   <PUBLISHER> Addison-Wesley </PUBLISHER>
   <YEAR> 1985 </YEAR>
</BOOK>
```

9/11/2005 11

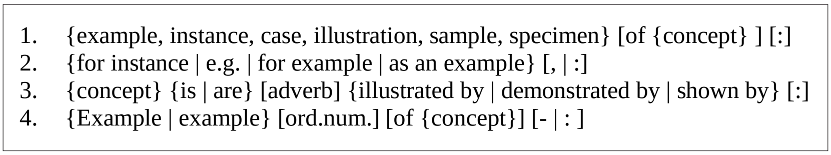
(b)



Besides this pattern-based approach, we apply the following simple heuristics to determine the instructional role of slides (COs of type *alocomcs:Slide*):

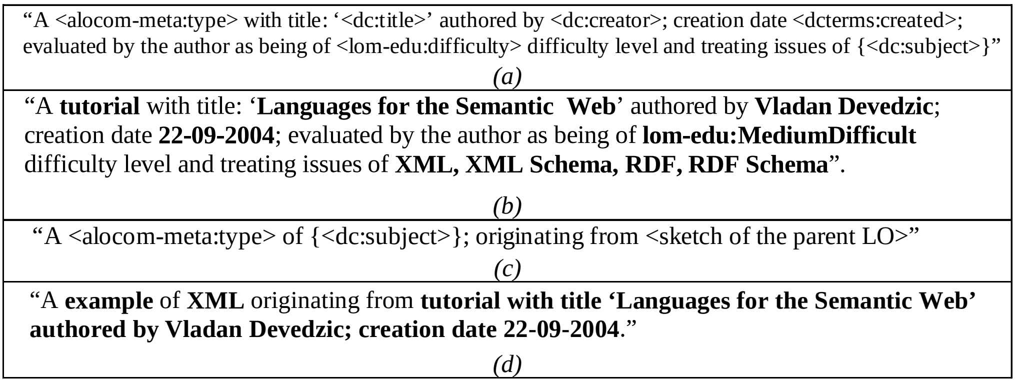
- If the title of a slide contains one of the following terms: "Content," "Outline," or "Overview," and the content of the slide's body is presented in the form of a list of items, the slide is assumed to have instructional role of the type *alocomct:Overview*. Similarly, if the title of a slide is "Summary" or "Conclusion," while the content of the slide's body is structured in the form of a

Figure 11. Patterns applied in TANGRAM for recognizing examples



1. {example, instance, case, illustration, sample, specimen} [of {concept}] [:]
2. {for instance | e.g. | for example | as an example} [, | :]
3. {concept} {is | are} [adverb] {illustrated by | demonstrated by | shown by} [:]
4. {Example | example} [ord.num.] [of {concept}] [- | :]

Figure 12. dc:description metadata element, templates (a, c), and examples (b, d)



"A <alocom-meta:type> with title: '<dc:title>' authored by <dc:creator>; creation date <dcterms:created>; evaluated by the author as being of <lom-edu:difficulty> difficulty level and treating issues of {<dc:subject>}"

(a)

"A **tutorial** with title: '**Languages for the Semantic Web**' authored by **Vladan Devedzic**; creation date **22-09-2004**; evaluated by the author as being of **lom-edu:MediumDifficult** difficulty level and treating issues of **XML, XML Schema, RDF, RDF Schema**".

(b)

"A <alocom-meta:type> of {<dc:subject>}; originating from <sketch of the parent LO>"

(c)

"A **example** of **XML** originating from **tutorial with title 'Languages for the Semantic Web' authored by Vladan Devedzic; creation date 22-09-2004**."

(d)

list, the *alocom-meta:type* metadata element of that slide is assigned a reference to the *alocomct:Summary* concept.

- If we can identify an alias of a certain domain concept(s) in the text of the slide's title, and the slide's body contains only an image (i.e., one or more CFs of the *alocomcs:Image* type), the slide is assumed to be an illustration of the domain concept(s) identified in the slide's title. Therefore, the slide is marked-up with *alocomct:Illustration* as its instructional role.
- If the content of the slide's title is one of the following terms/phrases: "Bibliography," "References," "Reference list," while the content of the slide's body is structured as a list, the instructional role of the slide is presumed to be of type *alocomct:Bibliography*. Additionally, each list item appearing in the slide's body is assumed to be of *alocomct:Reference* instructional type.

dc:description element

We generate a value for the *dc:description* metadata element of a content unit starting from the (inferred) values for other elements of its metadata set. This metadata is automatically generated both for COs and for LOs, that is, it is one of the metadata elements that are automatically generated even for LOs. Figure 12a shows the template used for generating a description of an LO, that is, a value for the LO's *dc:description* metadata element. Note that metadata elements appearing in the angled brackets in the template are replaced by their actual values. Curly brackets indicate that the enclosed element can have multiple values, as the example in Figure 12b illustrates. The figure presents automatically generated description for the LO from which originates the slide shown in Figure 10.

To generate a value for the *dc:description* element of a CO, we apply the template shown in Figure 12c. One should notice that the element "sketch of the parent LO" from the template refers to the concise version of the template presented in Figure 12a. More precisely, it is a part of an LO's description made according to the following template: "A <alocom-meta:type> with title: '<dc:title>' authored by <dc:creator>; creation date <dcterms:created>." Figure 12d shows the automatically generated value for *dc:decription* metadata element of the slide from Figure 10.

dc:identifier element

Each instance of the ALOCoM CS ontology that represents either an LO or a content unit of an LO (CF, CO, or one of their subclasses) is assigned a unique identifier. The identifier is stored in the form of the *dc:identifier* metadata. To generate a unique value for this metadata element, we use a modified version of the algorithm proposed in Vaucher and Ncho (2004). Figure 13 presents a method of the *tangram.utility.BasicUtilities* class that we use to generate content units' IDs.

Figure 13. Method for generating a unique value for a content unit's identifier

```
public class BasicUtilities {
  protected static int cidCounter = 0;
  …
  public String genComponentID(String cuType) {
     String cidBase = cuType + hashCode() + System.currentTimeMillis()%10000 + "_";
     return cidBase + (cidCounter++);
  }
  …
}
```

The top part of Figure 14 shows the OWL XML binding of the slide presented in Figure 10; the bottom part of the figure shows the slide's metadata. As the figure shows, a slide is related to its metadata set (an instance of the *alocom-meta:Metadata* class) via *alocom-core:metadata* property. Additionally, each slide is related with its components through *alocom-core:hasPart* and *alocom-core:ordering* properties, whereas *alocom-core:isPartOf* property is used to establish the relationship between the slide and its parent LO (i.e., slide presentation). The same formalism is used to store all other types of content units (and their metadata) in TANGRAM's LO repository.

EVALUATION

We did an evaluation of TANGRAM's annotation subsystem. Although limited in scope, the evaluation helped us identify strengths and weaknesses of the current solution. The evaluation was primarily focused on slides as content units that proved to be the

Figure 14. An instance of the alocomcs:Slide *class in XML syntax (a), and its metadata compliant to TANGRAM's LOM RDF Binding profile (b)*

```
<rdf:Description rdf:about="http://alocom/content-structure/slide.owl#Slide130074962749_356">
  <alocom-core:metadata rdf:resource="http://alocom/metadata.owl#meta-Slide130074962749_356"/>
  <alocom-core:hasPart rdf:resource="http://alocom/content-structure/slide.owl#Title130074962749_358"/>
  <alocom-core:hasPart rdf:resource="http://alocom/content-structure/slide.owl#SlideBody130074962749_357"/>
  <alocom-core:ordering rdf:nodeID="A333"/>
  <rdf:type rdf:resource="http://alocom/content-structure/slide.owl#Slide"/>
  <alocom-core:isPartOf rdf:resource="http://alocom/content-structure/slide.owl#SlidePres130074962368_0"/>
</rdf:Description>
```

(a)

```
<rdf:Description rdf:about="http://alocom/metadata.owl#meta-Slide130074962749_356">
  <dc:format rdf:resource="http://ltsc.ieee.org/2002/09/lom-educational#mime03"/>
  <dc:title>XML</dc:title>
  <lom-edu:difficulty rdf:resource="http://ltsc.ieee.org/2002/09/lom-educational#medium_difficult"/>
  <dc:subject>http://tangram/iis-domain.owl#xmltech01</dc:subject>
  <alocom-meta:type rdf:resource="http://alocom/content-model.owl#Example"/>
  <dc:identifier rdf:resource="http://alocom/metadata.owl#/id/Slide130074962749_356"/>
  <dc:creator rdf:nodeID="A98"/>
  <dc:language rdf:resource="http://ltsc.ieee.org/2002/09/lom-educational#lang01"/>
  <rdf:type rdf:resource="http://alocom/metadata.owl#alocom-metadata"/>
  <dc:description>I contain a example of XML originating from: A tutorial with title:"Languages for The
      Semantic Web" authored by Vladan Devedzic</dc:description>
</rdf:Description>
```

(b)

most reusable. Additionally, we were primarily interested in evaluating the precision of the techniques and heuristics applied for semantic markup of learning content, since semantic metadata proved to be the most relevant for automating content reuse. Accordingly, recognition of domain concept(s) and instructional role(s) of content units was the central point of the conducted evaluation. The evaluation consisted of the following two parts:

1. *Quantitative evaluation* using well-known information retrieval measures precision and recall and involving human subjects to specify the reference standard;
2. *Qualitative evaluation* involving human subjects to provide their comments to the recognized concepts in both types of ontology-based recognition.

Test Set

In both evaluations of TANGRAM's semantic annotation subsystem, we used a set of 54 slide presentations, all together consisting of 1,674 slides. The collected slide presentations were authored collected by the members of both the GOOD OLD AI Research Group of the University of Belgrade and the Laboratory for Ontological Research of Simon Fraser University. The analyzed slide presentations have been developed for different purposes such as teaching undergraduate and graduate courses, conference presentations, tutorials, and invited talks. All the slide presentations cover topics captured by the discussed domain ontology of IIS.

Quantitative Evaluation

Standard information retrieval evaluation measures, precision and recall have also widely been adopted by the Semantic Web community for evaluating different tasks such as semantic annotation (Cimiano, Ladwig, & Staab, 2005) and ontology alignment (Ehrig & Euzenat, 2005). In order to perform the evaluation using this approach, we first had to define a *reference standard* — a predefined model that is used to compare against the results of TANGRAM's annotation subsystem.

Reference Standard

A reference standard is usually defined by human experts. However, it is very hard to have a full agreement of domain experts upon different classification decisions (Calvo, Lee, & Li, 2004). In order to define as more confident reference standard as possible, we asked three human subjects to collaboratively annotate all slides from the sample with respect to the domain and ALOCOM CT ontologies. In fact, they had to make a consensual decision how each slide from the sample is to be annotated with respect to both the domain and ALOCoM CT ontologies.

Definition of Evaluation Measures

Given a number of answers in the reference standard ($|R|$), *precision (Pre)* is defined as the ratio of the number of correct answers ($|R \cap A|$) and total answers ($|A|$), while *recall (Rec)* is the ratio of the number of correct answers ($|R \cap A|$) and the number of answers defined in the reference standard ($|R|$). Formally speaking, they are defined as follows:

$$\text{Pre} = \frac{|\text{correct answers}|}{|\text{total answers}|} = \frac{|R \cap A|}{|A|} \qquad (1)$$

$$\text{Rec} = \frac{|\text{correct answers}|}{|\text{answers in reference standard}|} = \frac{|R \cap A|}{|R|} \qquad (2)$$

Findings

In Table 1 we give averaged values of precision and recall we obtained by annotating the analyzed test set using TANGRAM's semantic annotation subsystem. We chose to use *microaveraging* where average precision and recall are calculated by summing over all individual decisions for each specific slide (Sebastiani, 2002). That is to say, we consider as equal the annotation of each slide.

Table 1. Precision and recall for the analyzed test set

Type of annotation	Recall	Precision
Domain ontology	1 (1674/1674)	0.89 (1494/1674)
Pedagogical role	0.72 (1080/1503)	0.88 (1080/1224)

The reason why recall for COs' (i.e., slides') annotations with domain ontology concepts has the value 1 is that TANGRAM's algorithm for annotation (see the "Mining Metadata Values" section) attaches the domain ontology concept(s) of the parent LO (i.e., slide presentation) if no concept can be mined from the CO's content. Therefore, slides are always annotated with respect to the domain ontology. Of course, a slide typically covers a more or less narrower concept than its parent (i.e., concept related via *skos:narrower* property in the domain ontology; see the "Domain Ontology" section), or a concept that is in some other way semantically related to the one assigned to its parent (via *skos:related property* of the domain ontology). It may also happen that the topic of a slide (due to its specificity) is not included in the domain ontology at all. For example, a presentation might be annotated with the "KDD"[7] concept, while one of its slides might define "belief-driven evaluation" and how it is used to verify the relevancy of the knowledge acquired in a KDD process. However, as the domain ontology does not define a concept of "belief-driven evaluation," the TANGRAM's annotation subsystem is ignorant of this domain concept (all its domain knowledge comes from the employed domain ontology) and hence not able to determine the slide's semantic by mining the slide's content. As a result, the slide is annotated with the concept of "KDD" assigned to the slide's parent — not actually incorrect, rather not precise enough. However, as the standard evaluation measures can not distinguish between *fully correct answers* and *almost correct answers* (Ehrig & Euzenat, 2005), we decided to count such answers as correct when computing recall. Still, we thought it is important to see the influence of such *almost correct answers* on precision, so we used them as incorrect when computing precision. Actually, we had 1494 *fully correct answers* instead of 1674 correct answers used for calculating recall. The result was a bit lower precision (0.89) than recall, but still very competitive with the relevant Semantic Web annotators (Uren, Cimiano, Iria, Handschuh, Vargas-Vera, Motta, & Ciravegna, 2006). It should also be noted that the annotation with domain ontology concepts is rather influenced by the set of domain concepts that an LO author assigns to the LO when annotating it (during the upload procedure). The more precise that initial annotation is, the more chance that domain topics assigned to LO's components are satisfactorily precise. This also leads us to the conclusion that we should start exploring the use of more advanced text processing categorization techniques in order to avoid the big influence of the manually made annotations of LOs.

From Table 1 it is obvious that the value of recall is much lower for pedagogical role recognition than for domain concept recognition, as the manually submitted annotation of parent LOs (e.g., slide presentations) could not be applied to child COs (i.e., slides). This is due to the different content types that are allowed to be assigned to LOs and COs according to the ALOCoM CT ontology (shown in Figure 4). The value of recall shown in Table 1 can slightly be increased (0.75) if we consider the fact that title slides of slide presentations actually do not have a pedagogical role. Precision for pedagogical role recognition has almost the same value as precision of the domain ontology recognition. Precision of the pedagogical role is much higher than the average precision (0.6123) given in the paper (Liu et al., 2003) that was used as an inspiration for TANGRAM's pedagogical role annotation algorithm. This is probably due to the fact that authors of slide presentations use much less presentational patterns than the authors of many different types of Web resource on the Web. However, that shows that there is a lot of room for further evaluation of TANGRAM's approach in other domains (especially none computer science related) as well as on some other types of content units.

Note also that the most frequently occurring type of a slide is the one that explains a domain topic. It is not a definition in a literal sense, but somehow it defines the topic under discussion. The natural question raised: How to classify it? We decided to classify it as having definition as its pedagogical role, and that resulted in a large number of definitions. A further discussion about this issue is given in the next sub-section on qualitative evaluation.

Qualitative Evaluation

To evaluate the effectiveness of the patterns and heuristics we used for recognizing pedagogical role(s) of LOs' content units, we asked the authors of the slide presentations from the sample to determine the instructional role of each slide (s)he authored. Then we compared their responses with the values that TANGRAM automatically generated to describe instructional roles of the same slides. The system generally proved as satisfactorily effective, except for recognizing definitions. This can be explained by different comprehension of the semantics of the term *definition* among the interviewed content authors: it turned out that some of them had a strict, "mathematical" approach to this term, whereas others understood *definition* as any text that either formally or informally defines a concept from the subject domain. As we had the latter view in mind when formulating patterns for definition mining, it can be said that TANGRAM is well capable of recognizing such kind of content units.

Additionally, we used the same sample of slide presentations to determine how effectively TANGRAM infers the semantics of content units, that is, the concepts of the subject domain to which they refer. Again, the evaluation was based on a comparative analysis of the authors' and the system's "perception" of the semantics of the slides from the sample. We noticed that the system has a problem differentiating between two domain concepts if an alias of one concept is a part of an alias of another concept. For example, the domain concept with URI *iis:xmltech01* has "XML" as one of its aliases, whereas the concept with URI *iis:xmltech02* has an alias "XML Schema." When the system encounters a content unit comprising "XML Schema" phrase, it assigns both *iis:xmltech01* and *iis:xmltech02* concepts to the *dc:subject* metadata of the content unit. We are currently exploring how text mining techniques (e.g., part of speech taggers) can help us solve this problem. It is important to note that the algorithm for inferring the semantics of content units is completely ignorant (and independent) of the subject domain; all its knowledge comes from the applied domain ontology. Therefore, the same algorithm can be used to infer the semantics of any other domain, provided that a content ontology of that specific domain is available.

RELATED WORK

The KIM platform and framework provides a novel Knowledge and Information Management infrastructure and services for automatic semantic annotation, indexing, and retrieval of documents (Popov, Kiryakov, Kirilov, Manov, Ognyanoff, & Goranov, 2003). The platform is based on the PROTON ontology (http://proton.semanticweb.org), a light-weight upper-level ontology developed in the scope of the SEKT[8] project, as well as on two KIM specific ontologies: KIM System Ontology and KIM Lexical Ontology. Additionally, KIM is equipped with a Knowledge Base (KB) providing extensive coverage of entities of general importance. The platform comprises an infrastructure for information extraction and ontology-based annotation. This infrastructure is based on General Architecture for Text Engineering — GATE (http://gate.ac.uk), which has proved as a mature, extensible, and application-independent framework for information extraction and other natural language processing tasks. The advantage of our approach over the one on which KIM is based, is that we automatically generate values for a variety of metadata elements aimed at content markup, and not focus only on the subject matter of the content as KIM does. On the other hand, KIM applies much more elaborated text mining techniques than we do.

PiggyBank lets Web users extract individual information items from within Web pages and save them in a Semantic Web format (i.e., RDF), together with their metadata (Huynh, Mazzocchi, & Karger, 2005). The items, collected from different Web sites, can then be manipulated (browsed, searched, sorted, organized, etc.) together, regardless of their origins and types. On sites that do not publish RDF, PiggyBank can invoke screen-scrapers to restructure information found within their Web pages into RDF format. Semantic Bank is a repository of RDF triples to which a community of PiggyBank users can contribute and share the information they have collected. The core idea of PiggyBank is similar to ours: collect information resources from various sources, present them in an ontology-based format, and annotate them with metadata — the primary motivation is the need to re-purpose such information in order to cater the individual user's needs and preferences. Unlike PiggyBank, which targets Web pages and Web sites, we focus on the learning resources presented in the form of slide presentations. However, we plan to extend our system to other types of LOs in our future research.

Semi-automatic annotation of learning resources based on document layout features is proposed in Dehors, Faron-Zucker, Stromboni, and Giboin (2005). The approach presumes that each content author has a specific pedagogical approach that reflects on the structure and layout features of the documents he/she creates. The annotation task begins by interviewing the author of a document, in order to determine the relations between the employed presentational features and the envisioned educational approach. Subsequently, a phase of content re-authoring takes place to ensure that the employed visual features are compliant to the established instructional model. Only then it is possible to automatically identify and annotate content units according to their pedagogical role. The employed pedagogical ontology is generated on the fly and includes concepts that formalize elements of a content author's specific pedagogical strategy. Although this approach tends to be more precise in recognition of instructional roles of content units than the approach we propose, it is also more restrictive as it requires from content authors to strictly obey to the once established authoring styles. Additionally, it requires more human effort: interviewing the author and content re-authoring. Finally, learning resources are annotated only with their instructional roles, whereas we use a range of metadata elements to annotate them.

Automatic Metadata Generation (AMG) framework (Cardinaels, Meire, & Duval, 2005) is aimed for automatic annotation of LOs with metadata compliant to the IEEE LOM schema. Unlike our system, AMG does not enable semantic annotation of LOs — it cannot formally represent the semantics of LOs. Therefore, metadata that it generates are aimed only for human consumption — they cannot be comprehended and used by intelligent agents or any other piece of software.

Context-driven and Pattern-based ANnotation through Knowledge On the Web (C-PANKOW) is a method for automatic semantic annotation of Web content. The main idea herein is to approximate semantics by considering information about the statistical distribution of certain syntactic structures over the Web (Cimiano et al., 2005). The ambiguity, as an important problem in such an approach, is tackled by taking into account the context in which the entity to be annotated appears. C-

PANKOW applies much more elaborated techniques for content annotation than our approach suggests. However, it focuses only on identifying domain topics in analyzed documents, while we also aim at formally representing the structure of analyzed documents and mining the pedagogical role(s) of the content units comprising that structure. The common feature of the two approaches is that both are quite successful when focused on a specific domain (formalized in a domain ontology), while not that effective for domain-neutral annotations (when working with a general purpose ontology [i.e., WordNet, http://wordnet.princeton.edu/], C-PANKOW produces significantly poorer results [Cimiano et al., 2005]). Additionally, both are currently restricted to a specific document format: C-PANKOW focuses on HTML documents, whereas TANGRAM focuses on slide presentations.

KNOWITALL is an autonomous domain-independent system that automates the process of extracting large collections of facts from the Web (Etzioni, Cafarella, Downey, Kok, Popescu, Shaked, et al., 2004a). The only domain-specific input to KNOWITALL is a set of classes and relations to set its focus; no manually tagged training set is required. Information extraction is performed in two stages: (1) a set of domain-independent extraction patterns is used to generate candidate facts, (2) the plausibility of the candidate facts is evaluated using the pointwise mutual information (PMI)[9] measure. The authors have provided three extensions to the baseline system, namely rule learning, subclass extraction, and list extraction, hence improving the overall performance of the system (Etzioni, Cafarella, Downey, Kok, Popescu, Shaked, et al., 2004b). The approach of Etzioni et al. can be viewed as being orthogonal to ours: while we are concerned with annotating a given document with appropriate domain concepts, Etzioni et al. aim at learning the complete extension of a certain concept in order to build a search engine "knowing it all." On the other hand, we believe that their work on automatic learning of domain specific rules (i.e., patterns) can be equally well applied for solving one of the main challenges we are faced with — inferring pedagogical roles of content units originating from different domains and authoring styles.

The recent research from the knowledge capture field seems very relevant for the described approach. In Carenini, Ng, and Zwart (2005) the authors reported on their experience in extracting knowledge from evaluative text. They tried to employ WordNet for discovering similarities between a domain taxonomy and users' comments to some products written in plain text. A similar approach could be applied in TANGRAM when automatically annotating content units with respect to the domain ontology.

Finally, we briefly report what a recent comprehensive study on the present state of semantic annotation (Uren et al., 2006) has to say about automatic annotation. The study recognizes three general categories of automation approaches. The most common one uses manually written rules (patterns or wrappers), hence relying on the structure of documents (i.e., texts) for inferring proper mark-up. Our approach belongs to this category, as well as the previously mentioned KIM and PiggyBank systems. The other two kinds of systems apply diverse machine-learning approaches to learn how to annotate content. Supervised systems learn from sample sets of manually marked up documents. Their main disadvantage is that picking enough good examples is a non-trivial and error-prone task. Unsupervised systems (the third category) are starting to tackle this challenge by exploiting unsupervised learning techniques (e.g., C-PANKOW, KNOWITALL). Uren et al. also identify the present research challenges, among which relation extraction and annotation of multimedia documents (images, audio, video) are the most notable.

CONCLUSION

Aiming at reducing the cost of authoring high quality learning materials, we first developed an ontological foundation, called ALOCoM, for describing the structure and pedagogical role(s) of LOs and their components. Subsequently we extended that foundation by developing an approach aimed at automatic annotation of LOs and their content units. In this paper we present the developed approach that relies on both ALOCoM and domain ontologies. The general principles of ontology-based annotation of LOs' content units are implemented in TANGRAM, our learning environment for the domain of Intelligent Information Systems. Although the proposed approach is illustrated on a specific domain (Intelligent Information Systems), it is domain independent and can be applied to any other domain just by using another domain ontology. A brief description and demonstration of TANGRAM's functionalities as well as the ontologies referred to in the paper can be found at http://iis.fon.bg.ac.yu/TANGRAM/home.html

Our future work will be directed toward improving existing functionalities of TANGRAM's annotation subsystem and augmenting it with additional ones required for recognition of pedagogical roles not included in the current solution. Specifically, our intention is to empower TANGRAM with advanced features of the latest frameworks for natural language processing and information extraction tasks, such as: the already mentioned GATE and KIM (Popov et al., 2003) frameworks, as well as MontoLingua (http://web.media.mit.edu/~hugo/montylingua) — an end-to-end natural language processor for English with common sense. Furthermore, we plan to explore the potentials of learning designs and other formal educational modeling languages to serve as sources of context-related metadata for LOs and their content units. We believe that context-relevant metadata can be derived from descriptions of the learning processes in which LOs have actually been used or are intended to be used.

REFERENCES

1484.12.1 IEEE standard for learning object metadata. (2002, June). Retrieved October 15, 2005, from http://ltsc.ieee.org/wg12

Brusilovsky, P. (1998). *Adaptive hypertext and hypermedia*. The Netherlands: Kluwer Academic Publishers.

Calvo, R. A., Lee, J. M., & Li, X. (2004). Managing content with automatic document classification. *Journal of Digital Information*, *5*(2), Article No. 282. Retrieved December 20, 2005, from http://jodi.tamu.edu/Articles/v05/i02/Calvo/

Cardinaels, K., Meire, M., & Duval, E. (2005). Automating metadata generation: The simple indexing interface. In *Proceedings of the 14th International WWW Conference*, Chiba, Japan (pp. 548-556).

Carenini, G., Ng, R. T., & Zwart, E. (2005). Extracting knowledge from evaluative text. In *Proceedings of the 3rd International Conference on Knowledge Capture*, Banff, Canada (pp. 11-18).

Cimiano, P., Ladwig, G., & Staab, S. (2005). Gimme' the context: Context-driven automatic semantic annotation with CPANKOW. In *Proceedings of the 14th International WWW Conference*, Chiba, Japan (pp. 332-341).

Dehors, S., Faron-Zucker, C., Stromboni, J., & Giboin, A. (2005). Semi-automated semantic annotation of learning resources by identifying layout features. In *Proceedings of the International Workshop on Applications of Semantic Web Technologies for E-Learning*, Amsterdam, The Netherlands (pp. 65-69).

Dolog, P., Gavriloaie, R., Nejdl, W., & Brase, J. (2003). Integrating adaptive hypermedia techniques and open RDF-based environments. In *Proceedings of the 12th International WWW Conference*, Budapest, Hungary (pp. 88-98).

Duval, E., & Hodgins, W. (2003). A LOM research agenda. In *Proceedings of the 12th International WWW Conference*, Budapest, Hungary (pp. 1-9).

Ehrig, M., & Euzenat, E. (2005). Relaxed precision and recall for ontology matching. In *Proceedings of the KCAP2005 Workshop on Integrating Ontologies*, Banff, Canada (pp. 25-32).

Etzioni, O., Cafarella, M., Downey, D., Kok, S., Popescu, A.-M., Shaked, T., et al. (2004a).

Web-scale information extraction in KnowItAll (preliminary results). In *Proceedings of the 13th World Wide Web Conference*, Budapest, Hungary (pp. 100-109).

Etzioni, O., Cafarella, M., Downey, D., Kok, S., Popescu, A.-M., Shaked, T., et al. (2004b). Methods for domain-independent information extraction from the Web: An experimental comparison. In *Proceedings of the 19th AAAI National Conference on Artificial Intelligence*, San Jose, CA (pp. 391-398).

Felder, R., & Silverman, L. (1988). Learning and teaching styles in engineering education. *Journal of Engineering Education, 78*(7), 674-681.

Huynh, D., Mazzocchi, S., & Karger, D. (2005). PiggyBank: Experience the Semantic Web inside your Web browser. In *Proceedings of the 4th International Semantic Web Conference*, Galway, Ireland (pp. 413-430).

Jovanović, J., Gaševi, D., & Devedži, V. (2006). Dynamic assembly of personalized learning content on the Semantic Web. In *3rd European Semantic Web Conference*. Budva, Serbia & Montenegro (submitted).

Jovanovi, J., Gaševi, D., Verbert, K., & Duval, E. (2005). Ontology of learning object content structure. In *Proceedings of the 12th International Conference on Artificial Intelligence in Education*, Amsterdam, The Netherlands (pp. 322-329).

Liu, B., Chin, C. W., & Ng, H. T. (2003). Mining topic-specific concepts and definitions on the Web. In *Proceedings of the 12th International WWW Conference*, Budapest, Hungary (pp. 251-260).

Mohan, P., & Greer, J. (2003). Reusable learning objects: Current status and future directions. In *Proceedings of the 15th World Conference on Educational Multimedia, Hypermedia and Telecommunications*, Honolulu, HI (pp. 257-264).

Nilsson, M. (Ed.). (2002). *IEEE learning object metadata RDF binding*. Retrieved October 10, 2005, from http://kmr.nada.kth.se/el/ims/md-lomrdf.html

Popov, B., Kiryakov, A., Kirilov, A., Manov, D., Ognyanoff, D., & Goranov, M. (2003). KIM — Semantic annotation platform. In *Proceedings of the 2nd International Semantic Web Conference*, Florida (pp. 834-849).

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys, 34*(1), 1-47.

Sharable Content Object Reference Model (SCORM). (2004). Retrieved October 5, 2005, from http://www.adlnet.org/scorm/index.cfm

Stojanović, L. J., Staab, S., & Studer, R. (2001). E-learning based on the Semantic Web. In *Proceedings of the 6th World Conference on the WWW and the Internet*, Orlando, FL.

Ullrich, C. (2005). The learning-resource-type is dead, long live the learning-resource-type! *Learning Objects and Learning Designs, 1*(1), 7-15.

Uren, V., Cimiano, P., Iria, J., Handschuh, S., Vargas-Vera, M., Motta, E., & Ciravegna, F. (2006). Semantic annotation for knowledge management: Requirements and a survey of the state of the art. *Journal of Web Semantics, 4*(1), 14-28.

Vaucher, J., & Ncho, A. (2004). *JADE tutorial and primer*. Retrieved July 5, 2005, from http://www.iro.umontreal.ca /~vaucher/Agents/Jade/JadePrimer.html

Verbert, K., Jovanović, J., Gašević, D., & Duval, E. (2005). Repurposing learning object components. In *Proceedings of the OTM 2004 Workshop on Ontologies, Semantics and E-learning*, Agia Napa, Cyprus (pp. 1169-1178).

Verbert, K., Jovanović, J., Gašević, D., Duval, E., & Meire, M. (2005). Towards a global component architecture for learning objects: A slide presentation framework. In *Proceedings of the 17th World Conference on Educational Multimedia, Hypermedia and Telecommunications*, Montreal, Canada (pp. 1429-1436).

Verbert, K., Klerkx, J., Meire, M., Najjar, J., & Duval, E. (2004). Towards a global component architecture for learning objects: An ontology based approach. In *Proceedings of the OTM 2004 Workshop on Ontologies, Semantics and E-learning*, Agia Napa, Cyprus (pp. 713-722).

Wagner, E. (2002). Steps to creating a content strategy for your organization. *The E-Learning Developers' Journal*. Retrieved November 20, 2005, from http://www.elearningguild.com/pdf/2/102902MGT-H.pdf

Winter, M., Brooks, C., & Greer, J. (2005). Towards best practices for Semantic Web student modelling. In *Proceedings of the 12th International Conference on Artificial Intelligence in Education*, Amsterdam, The Netherlands.

ENDNOTES

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

[2] http://www.oasis-open.org/committees/dita

[3] http://www.adlnet.org/

[4] http://www.chiariglione.org/mpeg/standards/mpeg-21/mpeg-21.htm

[5] SKOS Core OWL binding is presented in Winter, Brooks, and Greer (2005): http://ai.usask.ca/mums/schemas/2005/01/27/skos-core-dl.owl

[6] The learner is more sequential in his/her learning style, hence tends to be confused/disoriented if the topics are not presented in a linear fashion (Felder & Silverman, 1988).

[7] KDD stands for Knowledge Discovery in Databases.

[8] SEKT (Semantically-Enabled Knowledge Technologies) — http://www.sekt-project.com/

[9] The PMI measure can be roughly defined as the ratio between the number of search engine hits obtained by querying with the discriminator phrase (e.g., "Liege is a city") by the number of hits obtained by querying with the extracted fact (e.g., "Liege").

Jelena Jovanović (http://iis.fon.bg.ac.yu/Jelena) received a BS and an MS in computer science from the Department of Computer Science, University of Belgrade, Serbia and Montenegro in 2003 and 2005, respectively. She is a PhD student and a teaching assistant at Department of Computer Science, University of Belgrade, Serbia and Montenegro. Her major research interests include Semantic Web technologies, learner modeling, adaptive learning environments, reusability of learning objects and learning designs, automatic metadata generation. She is currently pursuing her PhD thesis in the area of personalized learning on the Semantic Web. She is a member of the GOOD OLD AI research group.

Dragan Gašević (http://www.sfu.ca/~dgasevic) received his BS, MS and PhD in computer science from the Department of Computer Science, University of Belgrade, Serbia and Montenegro, in 2000, 2002, and 2004, respectively. He is a postdoctoral fellow at the Laboratory for Ontological Research, School of Interactive Arts and Technology, Simon Fraser University Surrey, Canada. His current research interests are in the area of ontologies, Semantic Web, integration between model driven engineering and ontology engineering techniques, and technology enhanced learning. So far, he has authored/co-authored more than 120 research papers, several book chapters, and two books. He has been severing on editorial and reviewing boards of several international journals. He has also been a program committee member and reviewer of several international conferences.

Vladan Devedžić (http://fon.fon.bg.ac.yu/~devedzic) received his BS, MS and PhD in computer science from the Department of Computer Science, University of Belgrade, Serbia and Montenegro, in 1982, 1988, and 1993, respectively. He is a professor of computer science at the Department of Information Systems and Technologies, FON - School of Business Administration, University of Belgrade, Serbia and Montenegro. 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 260 research papers, several book chapters and books. He has been severing on editorial and reviewing boards of several international journals. He has also been a chair, PC member, referee and for many international and conferences.