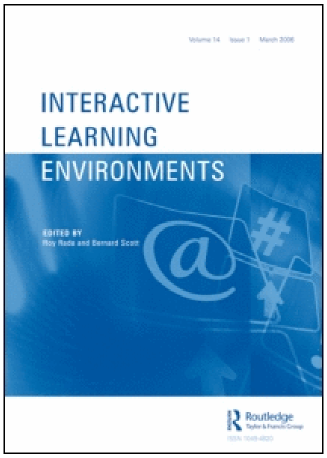

Interactive Learning Environments

Ontology-based annotation of learning object content

To cite this Article: , 'Ontology-based annotation of learning object content',
Interactive Learning Environments, 15:1, 1 - 26
To link to this article: DOI: 10.1080/10494820600968203
URL: http://dx.doi.org/10.1080/10494820600968203

Interactive Learning Environments
Vol. 15, No. 1, April 2007, pp. 1–26

Ontology-Based Annotation of Learning Object Content

Dragan Gašević[a]*, Jelena Jovanović[b], and Vladan Devedžić[b]
[a]*Athabasca University, Canada;* [b]*University of Belgrade, Serbia and Montenegro*

The paper proposes a framework for building ontology-aware learning object (LO) content. Previously ontologies were exclusively employed for enriching LOs' metadata. Although such an approach is useful, as it improves retrieval of relevant LOs from LO repositories, it does not enable one to reuse components of a LO, nor to incorporate an explicit specification of domain semantics into the LO content. We propose the use of domain ontologies to annotate LO content as well as content structure ontologies to enable direct access to LOs' components. That way, the same LO can be used in different ways and by different users, that is, it can be repurposed. In order to show the benefits of our proposal we discuss its application in adaptive learning systems. We also explore Semantic Web technologies and tools that are needed to support the presented approach.

1. Introduction

During the last couple of years the issue of learning objects (LOs) reusability persists as one of the most frequently discussed topics in the e-learning research community (Duval & Hodgins, 2003). Although the most widely adopted definition of LOs specifies a LO as "any entity, digital or nondigital that may be used for learning, education or training" (Duval, 2002), reusability has been mainly explored in terms of digital entities due to the huge influence of e-learning. There are several standards developed so far towards improving LOs reusability. For example, IEEE Learning Object Metadata—IEEE LOM (Duval, 2002) and Dublin Core (McClelland, 2003) are two initiatives specifying a standardized set of metadata for LO annotation. The ultimate goal of these standards is to facilitate search and retrieval of learning resources (i.e., LOs) and thus foster their reuse. For both of them, XML and RDF bindings are defined, so they can be used on the Web, and many developers have already based their repositories of LOs on top of those standards.

Recently, researchers have been proposing Semantic Web technologies, ontologies in particular, primarily for improving LOs' metadata (Berners-Lee, Hendler, & Lassila, 2001). For example, Mohan and Brooks (2003) have analysed relations of LOs and the Semantic Web, especially emphasizing the importance of ontologies.

*Corresponding author. School of Computing and Information Systems, Athabasca University, 1 University Drive, Athabasca, AB T9S 3A3, Canada. Email: dgasevic@acm.org

ISSN 1049-4820 (print)/ISSN 1744-5191 (online)/07/010001-26 © 2007 Taylor & Francis
DOI: 10.1080/10494820600968203

Accordingly, they identified several kinds of ontologies related to LOs: ontologies of domain concepts, ontologies formally specifying teaching and learning strategies, and ontologies about physical structuring of LOs. In the paper by Brase and Nejdl (2004), the authors give an example of an ontology developed in accordance with the ACM Computing Classification System (ACM CCS), and used in the Edutella P2P network. However, none of the existing solutions enable one to reuse just some specific parts of a LO or to repurpose a LO (i.e., use the same LO in different ways).

In order to address these open issues we advocate using ontologies to describe a LO's content in detail, thus providing the LO with a new dimension of reusability—content reusability. Our starting point is the classification of ontologies for e-learning given in the paper by Stojanović, Staab, and Studer (2001) that distinguishes between the following three types of ontologies:

- *Content* (domain) ontologies enabling one to formally state what the learning material is about;
- *Context* ontologies providing means to formally state in which form the learning content is presented;
- *Structure* ontologies formalizing the structure of the learning material.

Our focus is on content and structure ontologies, and we suggest using these ontologies to enable reuse and repurposing of both LOs (in their entirety) and their components. The main peculiarity of our approach compared to previous ones is that we emphasize ontology-based structuring of a LO and semantic annotation of both the LO as a whole and its components. Specifically, we argue for explicitly structuring a LO in accordance with a structure ontology and thoroughly annotating its content with concepts of one or more domain ontologies. Although, even enrichment of LOs' metadata with ontological concepts (as suggested in, e.g., Brase & Nejdl, 2004) improves the effectiveness of LOs retrieval, we are convinced that ontology-based structuring and annotation of LOs' content can further enhance it. That means, semantically organized LO's content has better potential to be repurposed. Although intuitively clear and straightforward, the suggested approach is technically very demanding. Therefore, we discuss some practical aspects of authoring reusable LO content, emphasizing the role of ontologies, necessary tools (specifically annotation tools and domain tools), and transformations.

The principle of having a common model (i.e., LO content) that can be repurposed (i.e., used in many different ways) resembles the Observer design pattern (Gamma, Helm, Johnson, & Vlissides, 1995)—widely known as the Model-View-Controller (MVC) pattern. According to this pattern the common model is observed by many observers; every time the model is changed, each observer is notified about the change in order to update its view. In this context, observers can be thought of as, say, different graphical representations of the same content (e.g., the data from an MS Excel table can be presented in the form of diagrams or histograms).

The suggested approach is also relevant from the aspect of learning content personalization (Jovanović, Gašević, & Devedžić, 2006). 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 preferences, objectives, competencies and/or other specific features of a learner relevant for the learning process. To exemplify this statement we discuss the benefits of ontologically supported LOs in terms of adaptive learning systems.

In the next section we provide an overview of the types of ontologies upon which our approach is based. Section 3 explains the role of these ontologies in the process of authoring semantically enhanced LOs' content and provides an insight into the present Semantic Web authoring and annotation tools. This section also explains how such LOs can be adapted to suit the learners' needs, that is, how they are used in an adaptive learning environment. In section 4 we give a concise overview of Semantic Web technologies supporting the presented approach. Section 5 sketches two application examples of the proposed approach. Section 6 reports on some related work, while section 7 concludes the paper with some directions for future research.

The paper uses terms and concepts from the Semantic Web domain that are well known to the Semantic Web community. We refer readers who might want to see the definitions of these terms and concepts to the Appendix.

2. Ontologies for Learning Objects

In this section, we try to clarify some aspects of the ontologies that we identified as important for achieving LOs' reusability.

Figure 1 depicts the kinds of ontologies that we find relevant for making the LO's content reusable: metadata ontology (MO), content structure ontology (CSO), and content or domain ontology (DO_i). The term metadata ontology refers to an ontological representation of any kind of metadata schema aimed at describing LOs with metadata. An example of such an ontology is the RDF binding of the IEEE LOM standard (Nilsson, Palmer, & Brase, 2003), as well as any application profile of

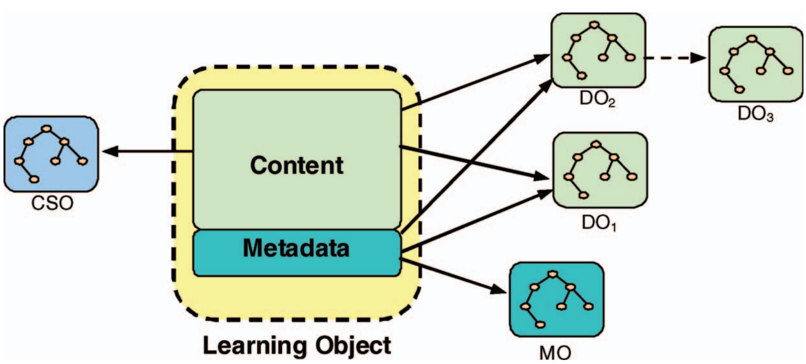

Figure 1. Ontologies describing learning objects: (a) metadata ontology (MO), (b) domain ontologies (DO_i), and (c) content structure ontology (CSO)

this binding. The other two kinds of ontologies are defined in Stojanovic et al. (2001) as being essential for the e-learning domain: The first is intended for explicitly defying the structure of a LO, whereas the second is aimed at formalizing shared conceptualization of a specific domain. Our approach assumes structuring a LO's content in accordance with a CSO, structuring its metadata in compliance with a MO, and using one or more DO to introduce explicit semantics into both LO content and metadata.

2.1 Content Structure Ontologies

An explicit definition of a LO structure is useful when one needs to reuse specific parts of a LO rather than the LO as a whole. In such situations, current practice is to copy and paste those parts of a document (e.g., a paragraph, a sentence, and an illustration) that are relevant for the situation at hand. However, this can be a rather tedious and time-consuming task. More importantly, such an approach, with the manual intervention it implies, does not scale well, as it does not allow for (semi-) automated processes to assist or take over completely. Content authors would be in a much better position if access to the components of LOs and their composition into meaningful units are made, at least partially, automatic. This can be accomplished by explicitly representing the structure of LOs, thus enabling reuse of their components as well. Explicitly defined structure of a LO allows for the disaggregation of a LO, that is, splitting it into its constituent content units. These content units, enriched with metadata, are thus made searchable and reusable on their own.

Since an ontology is, by definition, an explicit representation of a shared knowledge of a domain (Gruber, 1993), it is a straightforward solution for explicitly defining the structure of a LO. Such an ontology describes different types of the LO's components as well as the structure-related relationships between the components. It serves as a solid foundation for tools aimed at decomposing present LOs into ontology-defined components and vice-versa (ontology-defined LOs' components into a new LO).

In our previous collaborative research efforts with the ARIADNE research group from the Katholieke Universiteit Leuven, Belgium, we have developed the ALOCoM ontology as a content structure ontology based on the Abstract Learning Object Content Model (ALOCoM) (Verbert, Klerkx, Meire, Najjar, & Duval, 2004) and IBM's Darwin Information Typing Architecture (DITA) (Priestley, 2001). The ontology defines concepts and relationships that enable formal definition of a LO's structure. To learn more about this ontology, interested readers are referred to Jovanović, Gašević, and Devedžić (2005).

2.2 Content or Domain Ontologies

Nevertheless, explicit structuring of learning resources (based on a content structure ontology) is not enough for effective content reuse. We argue for further enhancing learning resources by providing domain ontology-based descriptions of their content, or more precisely, by adding pointers to the concepts of appropriate domain (content) ontologies (Devedžić, 2004). A domain ontology describes content of a LO in terms of concepts of the subject domain and their relationships. In Figure 1 domain ontologies are denoted as DO_1, DO_2, etc. Domain ontologies are further discussed in section 5.

Hence, we suggest creating those annotations on top of the learning content organized in accordance with a content structure ontology. A LO created using this principle gets a new dimension of reusability—it can be used in different ways within the same course. This is important in computer science courses like, for example, object-oriented modelling with Unified Modelling Language (UML). A teacher might use an UML model in a PowerPoint presentation, while students can try the same model in a CASE tool (e.g., IBM Rational Rose). The same principle can be applied in other disciplines (e.g., philosophy, history) as well. Furthermore, created this way LOs are more suitable for retrieval since their content can be inspected using ontology-based conceptualization.

3. An Ontology-Based Framework for Content Repurposing

In order to enable effective reuse and repurposing of LOs, we have to further enhance semantics of their content. Here, we recommend an approach that builds upon the traditional LO creational schema, but further extends it to incorporate support for semantic structuring and markup of LO content using content structure and domain ontologies, respectively. Figure 2 depicts the proposed approach for both developing and consuming semantics-aware LOs in an adaptive learning environment.

The central part of the figure is a repository of LOs holding semantics-aware LOs. LOs' metadata are presented in accordance with a metadata ontology (MO) (e.g., IEEE LOM RDF binding). This metadata can be enriched with the concepts from the domain ontologies (DO_i) (Brase & Nejdl, 2004). Additionally, each LO is assigned an ontology-based description of its structure (i.e., a CSO-compliant description), whereas its content is enriched with references to the concepts of one or more domain ontologies (DO_i).

However, the inclusion of ontology-grounded descriptions of a LO's content in its markup data should not impose additional burden on LO authors. On the contrary, we argue that authors should be ignorant of the existence of domain ontologies, since one cannot expect that, say, a teacher of social sciences can be aware of the technical role that ontologies play in knowledge engineering. In order to overcome this problem, we recommend either usage of existing (i.e., annotation tools) or development of new tools (see the next subsection) providing a graphical user interface for seamless creation of annotations (Handschuh, Volz, & Staab, 2003).

Later, the teacher uses these LOs within a course-authoring tool, when building courses. Thus, the process of composing a course is primarily based on searching for adequate LOs and "gluing" the retrieved LOs together according to an instructional model. Rich semantic descriptions of LOs make the search process both easier and more accurate. This way created learning courses are used in Adaptive Learning

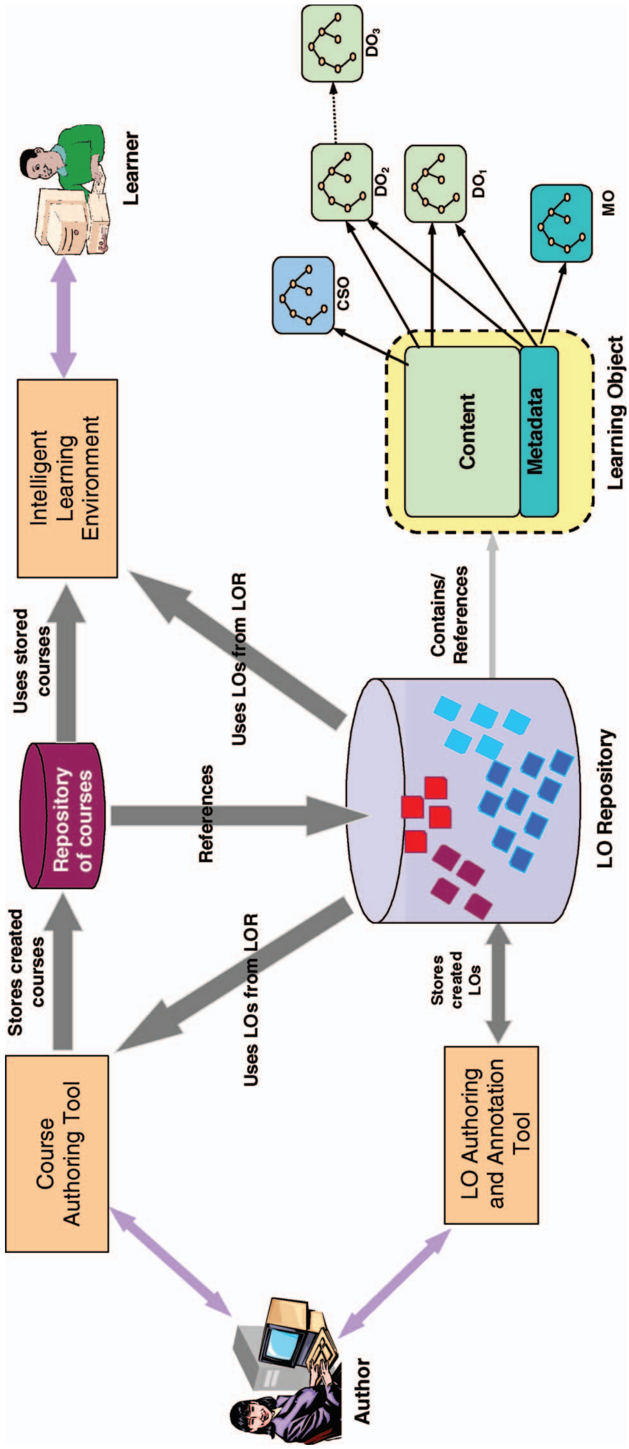

Figure 2. Extended LO creational schema

Environments (ALEs), where they become the subject of further transformations aimed at making them compliant with specific learning needs of each learner.

3.1 Learning Object Authoring and Annotations Tools

An essential prerequisite for establishing an ontology-based framework aimed at LO content reuse is the availability of tools that allow content authors to seamlessly compose LOs with semantically marked up content. Since widely accepted and well-known authoring tools (e.g., text processors, MS PowerPoint, HTML editors) do not provide the full support, we suggest employing either additional tools in the form of annotation tools or special domain tools that incorporate both authoring and annotation work.

3.1.1 Annotation tools. Annotation tools have started appearing as the result of a Semantic Web effort aimed at producing semantically marked up Web resources. The primary motive is to make the process of generating semantic descriptions less burdensome and tedious, therefore stimulating otherwise reluctant content authors to produce annotated resources (LOs in this case). The initial efforts were focused on creating specialized tools that support association of semantic markup with preexisting documents. However, this approach did not prove an efficient one and was not accepted in content authors' communities, since it exerted extra efforts and imposed an additional burden on content authors. As a result, we are currently facing an augmenting number of research efforts aimed at fully automating the annotation process. Actually, automatic semantic annotation (i.e., annotation with concepts from ontologies and/or classification schemes) is presently one of the hottest research topics in the Semantic Web community. A recent comprehensive study on the present state of semantic annotation (Uren et al., 2006) 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 markup. 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. 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.

The aforementioned encouraging developments in the field of automatic content annotation foster the development of advanced authoring tools, that is, tools that enable a content author (e.g., a teacher) to seamlessly create semantically enriched LOs. Ideally, such a tool would, in the background, perform the required semantic annotation of the LO under construction (i.e., insert pointers to appropriate ontology concepts). For example, Magpie is a plug-in for standard Web browsers (e.g., MS Internet Explorer and Mozilla FireFox) aimed at semantic interpretation of Web pages (Dzbor, Motta, & Dominque, 2004). A Magpie-supported browser uses different colours to highlight instances of different types of concepts from the domain ontology selected by the user. Even though Magpie is not an authoring tool, it can still facilitate the process of learning content authoring (e.g., course authoring) by drawing the teacher's attention to the relevant segments of the currently presented Web page (i.e., segments discussing relevant domain topics). Those relevant parts can subsequently be automatically extracted to be included in the course the teacher is authoring.

A complementary approach is the one applied in the MemoNote tool (Azouaou & Desmoulins, in press), a semantic annotation tool that enables teachers to seamlessly annotate digital documents with their own comments. The tool provides a teacher with a set of "annotation patterns", where each pattern is visually represented with a specific annotation form (i.e., colour used for highlighting the content) and has an associated semantics. For example, red highlighting can mean an important section to be used during the course authoring (i.e., to be extracted and plugged into the course structure). In fact, this idea has analogy in marking printed books and papers using a pen or marker. While reading a printed text, a teacher, uses these marks as reminders to the parts that (s)he found interesting for her or his course. An advantage that digital resources have is that denoted (i.e., annotated) parts can be automatically extracted.

Recently, a significant amount of annotation-related research efforts have started to focus on the development of integrated authoring and annotation systems. The ultimate goal is to provide authors with a single point of access systems that place the annotation process in the context of the authors' every day activities (Uren et al., 2006). One of the most notable research results is the AktiveDoc tool that supports sharing and reuse of knowledge during the document creation (writing) and use (reading) (Lanfranchi, Ciravegna, & Petrelli, 2005). It leverages Semantic Web technologies to support the production of ontology-based annotations while the document is written. Actually, three kinds of annotations are enabled: ontology-based content annotation, insertion of free-text (i.e., unstructured) comments, and on-demand document enrichment. Furthermore, the system monitors the user's actions while he or she is writing or editing a document and provides automatic suggestions about relevant content, hence enabling timely reuse of existing knowledge when available.

Figure 3 gives a comparative overview of the main characteristics of four well-known authoring and annotation tools: Ont-O-Mat—an implementation of the CREAM framework (Handschuh & Staab, 2002); SMORE (Kalyanpur, Hendler, Parsia, & Golbeck, 2003); Semantic Word (Tallis, 2003); and AktiveDoc (Lanfranchi et al., 2005).

It is also worth noting that presently available annotation tools support annotation of a limited number of document formats: for instance, HTML pages, Scalable Vector Graphics (SVG), and MathML document formats, as well as MS Word and MS PowerPoint documents. However, one can expect that the future tools will implement the support for other important formats of Web resources, such as PDF, Synchronized Multimedia Integration Language (SMIL), and different multimedia formats (e.g., for representing animation and sound).

Annotation tool	**Ont-O-Mat**	**SemanticWord**	**SMORE**	**AktiveDoc**
Document Format	HTML pages	MS Word documents	HTML pages, images and e-mail messages.	HTML documents
Annotation Form	Document's annotation is viewed as a set of instances of DAML+OIL classes, datatype properties (*attributes*) and object properties (*relationships*).	Two types of annotations: 1) *instance references* – associates a text region with an instance of a class; 2) *triple bags* – describes a text region with a collection of DAML+OIL subject-predicate-object triples.	Annotations are presented as standard RDF triples with the underlying subject-predicate-object model.	RDF triplets
Manual vs. automatic annotation	Initially manual; the upgraded version (S-CREAM) supports semi-automatic annot.	Semi-automatic	Manual	Semi-automatic
User support facilities	Web browser & editor; drag & drop based markup process; the upgraded version: suggestions for content annotation	Familiar authoring environment; extensible library of templates with partially annotated text	Web browser & editor; e-mail editor; ontology browser	Point-and-click annotation; suggestions for content annotation and potentially relevant content for inclusion; provision of semantic services
Storage of generated annotations	Both inside the original document and in the Annotation Inference Server.	Annotations are a part of the original document	Annotations are stored in a separate file; in case of e-mail messages that file is sent with the e-mail as an attachment.	Annotations are stored separately in a MySQL database.
Homepage	http://annotation.semanticweb.org/	http://mr.teknowledge.com/daml/SemanticWord/	http://www.mindswap.org/~aditkal/editor2.shtml	http://nlp.shef.ac.uk/wig/aktivedoc.htm

Figure 3. Summary view of main features of four widely known annotation and authoring tools

3.1.2 Domain tools. As we have already mentioned, the latest annotation tools can serve as authoring tools as well. However, their focus is restricted to general domains, like creation of semantically annotated HTML pages or Word documents. More specific domain tools (e.g., IBM Rational Rose in software engineering and AutoCAD in civil engineering) currently are not extended with support for content annotation. However, to better support learning in those specific domains, the domain tools need to be augmented with the ability to annotate artefacts they generate. For example, suppose that an expert in the field of software engineering uses a domain tool augmented with annotation capabilities to build UML models for the application that (s)he is developing. The created models (can be used for teaching or learning UML modelling, and thus we treat them as LOs) would be semantically described with concepts from the domain ontologies. Since semantically marked up LOs can be retrieved from the Semantic Web, a teacher of software engineering might use them to prepare a course on UML-based object-oriented modelling: For example, (s)he can incorporate those UML models into slides of a PowerPoint presentation (s)he is preparing.

3.2 The Role of Course Authoring Tool

Once created, an ontology-enhanced LO can be included in different courses. The principal role of a course authoring tool (CAT) is to enable seamless composition of new courses out of existing LOs available in the LO repository (LOR). Furthermore, it is supposed to enable an author to compose her or his own course model that suits his or her pedagogical approach the best. Our view of the architecture of such a tool is depicted in Figure 4. As the figure suggests, we envisage a tool of modular architecture with a coordination module (*Coordinator*) acting as an intermediary in communication between other CAT's modules.

As we stated, a CAT has to facilitate the search and retrieval of LOs available in the LOR. In the proposed architecture this search is based on concepts from domain ontologies (DO_i), as it is a straightforward way to find relevant LOs for any subject domain. The *Search Module* of the CAT uses a Semantic Web query language, such as RDF Query Language (RDQL) (Seaborne, 2004) or OWL Query Language (OWL-QL) (Fikes, Hayes, & Horrocks, 2003). However, one should keep in mind that the majority of course authors are neither familiar with the concept of ontology nor with query languages (especially those for the Semantic Web). Therefore, the CAT comprises a rich *User Interface Module* enabling users to complete the job of searching for LOs with just point-and-click and drag-and-drop actions, accompanied with minimum typing inputs.

When an author, using the CAT's search mechanism, finds a useful LO, (s)he can retrieve it and incorporate it into the instructional model of a course (s)he is creating. Here we consider an instructional model as a generic model of a course, module, or lesson created in accordance with an instruction (learning) design modelling language, such as the IMS Learning Design (LD) Specification (IMS LD, 2003). The resulting instructional model is presented in accordance with the propositions of

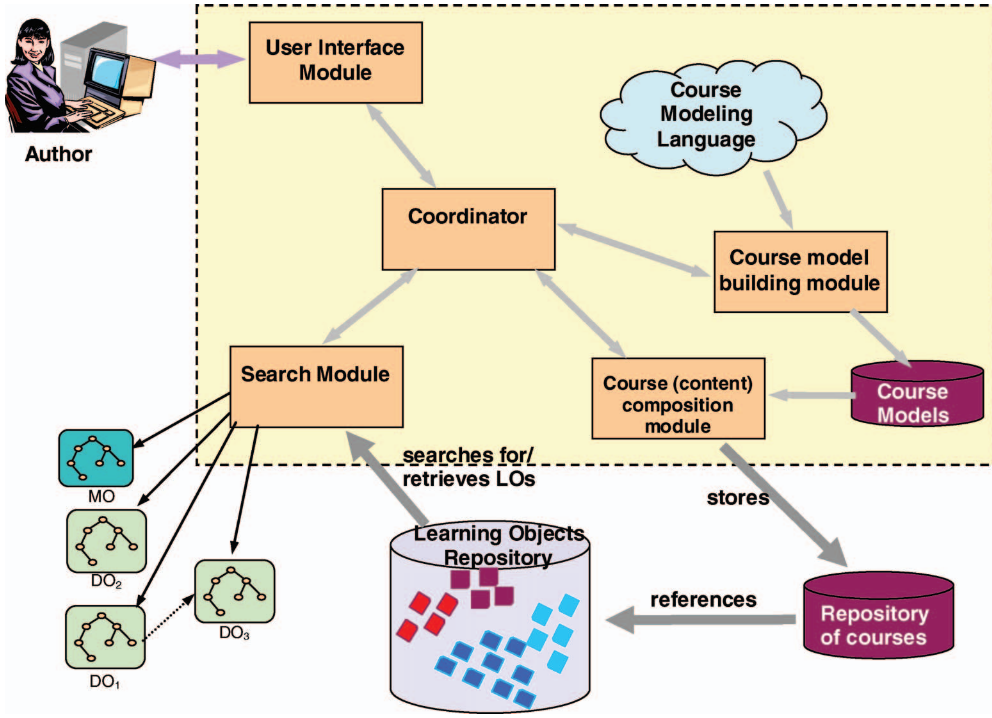

Figure 4. Architecture of the course authoring tool

the used modelling language (e.g., as an IMS LD compliant Unit of Learning), and then stored in the repository of courses. Inside the ALE, this course will be further transformed into a learner-suitable course, both in terms of its content and presentational form (see the next subsection). The CAT also facilitates semi-automatic annotation of the course under construction. The course description is automatically composed out of semantic descriptions of its constituent LOs, as it was proposed in Keenoy et al. (2004). Thus, ontology grounded semantic descriptions of LOs and their content play the major role in the semantic markup of the generated courses.

The course authoring tool also provides the author with an option of creating a new course model employing constructs of the supported learning design modelling language (e.g., IMS LD). A special module, which we named *Course Model Building Module*, in the CAT architecture supports this functionality.

3.3 Adaptation to Learner Profiles

Courses created using the previously described course authoring tool can be characterized as nonadapted in the sense that they are not customized to the needs of individual learners. Therefore we suggest an ALE that adapts these courses to the learning objectives, preferences, and knowledge background of each individual learner. Figure 5 illustrates our vision of the ALE's architecture.

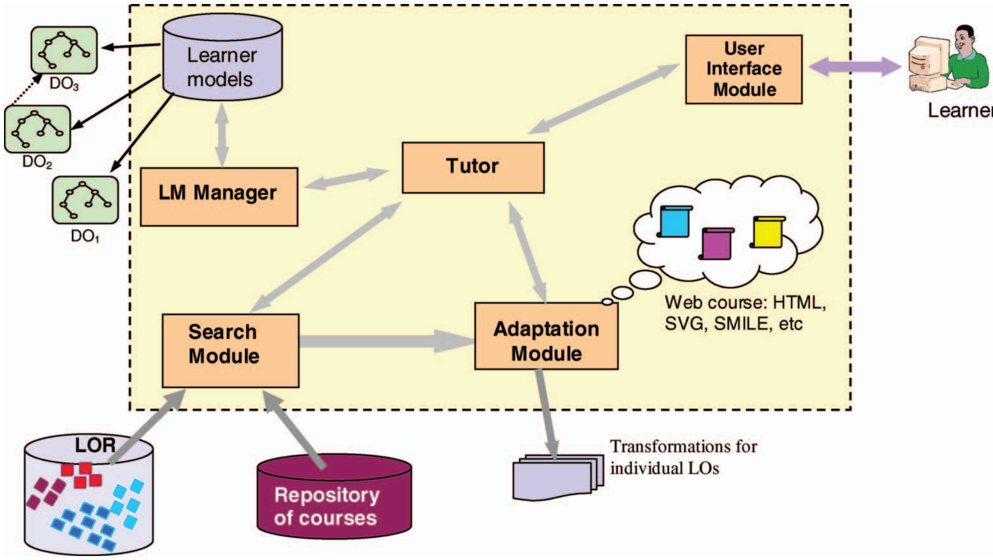

Figure 5. The architecture of an adaptive learning environment

A learner communicates with the ALE through its *User Interface Module*. The first thing that (s)he has to do is to select a course to work on. Since it is not such an easy task when there are a plethora of available courses, the system gives the learner a helping hand. More precisely, the learner specifies the topic (s)he is interested in, selecting a domain ontology and its concept(s) that best represents the topic, while the system does the rest. Actually, the system uses data stored in the learner's profile in order to formulate a more refined query for searching for (an) appropriate course(s) among the course descriptions stored in the repository of courses, and retrieves those that seem to fit the learner's current educational needs the best. The most important data for building those queries are:

- Learner's objectives—it seems a widely accepted approach to express learning objectives as targeted competencies, that is, what a learner will be able to do as a result of taking a certain course (Paquette, 2003). Although not formally accepted as the standard, Bloom's taxonomy of educational objectives (Bloom & Krathwohl, 1956) (defining, for example, knowledge, analysis, and application) is commonly used for this purpose (Keenoy, Levene, & Peterson, 2003).
- Previous learning experiences in related fields—the knowledge level of the topics that, according to the relevant domain ontology, constitute the set of prerequisites for the topic of interest.
- Learner's preferences regarding the pedagogical approach (instructional method) implemented in the course structure (e.g., active and exploration-based).

Having finished the search, the system provides the learner with a list of no more than four or five courses with a brief description of each one. The learner makes the final choice and only then the system pulls the chosen course from the URI provided in its description.

The course undergoes a transformation in order to provide the learner with the most suitable learning experience. As each course is composed of LOs, when transforming a course, we use transformations for all included LOs. Accordingly, transformations of the course depend on LOs' transformations. In this way, the same course can repurpose content of its constituent LOs in different ways. These transformations can, for example, mean content extraction, so that the ALE does not show the full LO's content, but only parts suitable for a concrete situation. This possibility is especially useful for tailoring the course's material according to the learner's previous knowledge of the subject domain: Beginners are introduced only to the basics of each domain concept, intermediary level learners are given more detailed information on the subject, while advanced learners are presented with the most complex aspects of the topic. The system is able to follow the progress of the learner and update his or her profile as (s)he makes headway through the course. It adapts the presented material to reflect the change in the learner's knowledge, so that the course always stays in accordance with the user's needs. Therefore, we introduced the *Tutor* module in the proposed architecture as a module that communicates with the *Learner Model (LM) Manager* and the *Adaptation Module* and coordinates their work in order to make permanent user adaptation possible. Furthermore, the learner's preferences related to the presentation of the course material is also taken into account. Different learners prefer different presentation formats: Some are fond of graphically presented contents, others prefer short video clips, while the third are used to the typical Web page format. Depending on this criterion selection of the learning content is made.

The aforementioned transformations are primarily based on methods and techniques established in the field of Adaptive Hypermedia. There are two basic adaptation methods: adaptive presentation (i.e., content level adaptation), and adaptive navigation support (i.e., link level adaptation) (Brusilovsky, 2001). For example, the link annotation technique together with the link hiding technique are used to implicitly instruct a learner as to which units of a course (s)he is ready for, which are too simple for his or her level of knowledge, as well as which should be (currently) avoided due to their complexity.

4. Required Semantic Web Technologies

This section gives an overview of technologies needed to support semantic annotation of LOs content (i.e., annotations with the concepts of one or more domain ontologies).

4.1 Domain Ontologies

Presently there are a plethora of specific domain ontologies already available on the Web. Actually, we can talk of libraries of ontologies created through contributions of the Semantic Web community members. For example, the DAML Ontology Library (http://www.daml.org/ontologies/) holds nearly 300 ontologies written in the DAML ontology language. Furthermore, Swoogle (http://swoogle.umbc.edu/), a crawler-based indexing and retrieval system for Semantic Web Documents (i.e., Web documents written in RDF or OWL ontology languages) offers a constantly increasing library of ontologies. In the time of writing this paper, Swoogle claims to be searching over 10,000 ontologies. Additional sources of ontologies are, for example, OwlSeek (http://www.owlseek.com/) providing an index of OWL ontologies, and SchemaWeb (http://www.schemaweb.info/), a large repository of ontologies expressed in the RDFS, OWL, and DAML + OIL languages. An author of a LO can search one of those libraries in order to find an ontology that best describes the content of the LO (s)he is creating. The authoring tool is supposed to provide this search facility, as well as a suitable graphical user interface which provides authors with intuitive visualization of the chosen ontology (e.g., a navigational tree).

An author can also be provided with a means to easily create an ontology (one or more of them) to be used for annotating the LOs (s)he develops. Since fully manual ontology modelling is a very difficult, error prone, and time consuming task, appropriate tool support is absolutely necessary. Actually, in the recent years, the Semantic Web community has been showing a constantly increasing interest in automating the process of ontology development—also known as ontology learning (Maedche & Staab, 2001). Ontology learning is a brand new research discipline that relies on and integrates achievements of a number of complementary disciplines, primarily computational linguistics, information retrieval, machine learning, databases, and software engineering. However, fully automatic ontology development is still in the distant future, and the current research efforts are directed towards enabling semi-automatic ontology development. In other words, the use of machine learning and natural language processing methods and techniques enable automatic knowledge acquisition from different kinds of unstructured (e.g., free text), semi-structured (e.g., HTML), and fully structured (e.g., databases) data, but human intervention is still needed to supervise the process and refine the results. Currently available tools, such as Text-To-Onto (a part of the KAON Semantic Web tool suite, http://kaon.semanticweb.org/), are primarily aimed for ontological engineers (i.e., experts in the field of ontology development), and are not that convenient for ontology-ignorant LO authors.

It cannot be denied that the idea of enabling authors to seamlessly make their own ontologies, that is ontologies that best suit their needs, sounds appealing. Nonetheless, there is one significant hindrance for wide acceptance of that kind of approach and it is related to the problem of enabling automatic mappings between different ontologies. How to enable autonomous agents on the Semantic Web to understand that two or more (differently named) concepts from different ontologies denote the same thing is still an open question. For example, in the context of e-learning a teacher might, using advanced features of a domain tool, construct a new ontology for the domain of his or her professional interests (e.g., the course(s) (s)he is teaching) as (s)he perceives that domain. Accordingly, the content of a LO the teacher creates is annotated with concepts of the teacher's proprietary ontology.

If another teacher from another university has applied the same procedure while creating his or her LO on the same subject, we would have two LOs with highly similar content, but differently annotated and autonomous agents have trouble grasping that those LOs treat the same subject. In fact, this issue of bridging various ontologies has already been recognized as a separate research area in the Semantic Web community called ontology mapping. The main task of ontology mapping is (semi-)automatic mapping discovery that can be formulated as follows (Noy, 2004): Given two ontologies, how do we find similarities between them, determine which concepts and properties represent similar notions, and so on. Dealing with this problem, researchers have been using various techniques based on different approaches such as machine learning, upper-level ontology, string matching, computational linguistics, graph matching, and model-based mapping (Shvaiko & Euzenat, 2005). Although the most recent experiments demonstrate encouraging results, there is still a lot of room for future improvements and research (Euzenat, Stuckenschmidt, & Yatskevich, 2005).

4.2 Semantic Web Query Languages

Reusability of LOs on the Semantic Web vastly depends on query languages that are used for searching LORs and retrieving LOs. Those languages must be (1) highly expressive, that is, they are ought to enable formulation of precise queries over LORs, and (2) not computationally intensive, that is, they should not pose too much computational burden on the part of the LOs providers (i.e., servers that host LORs). Currently, there are a few languages of this type: RQL (Karvounarakis, Alexaki, Christophides, Plexousakis, & Scholl, 2002), RDQL (Seaborne, 2004), TRIPLE (Sintek & Decker, 2002), and OWL-QL (Fikes et al., 2003), just to name the most important ones. It should be noted that these languages are not meant just for querying the Semantic Web, but also for performing transformations of the knowledge (content) expressed using ontological languages. Furthermore, they can be used to enable mappings between ontologies, as well as for defining views on ontologies. The main problem with those languages is that there is not a general consent which language should be accepted as the standard. A potential candidate is the recent W3C initiative for a standard transformation and query language based on OWL – OWL-QL.

Although access to LORs is facilitated by Semantic Web query languages, formulation of effective queries over LO metadata with complex organization (i.e., underlying schema) requires profound understanding of the schema that is often beyond the needs of an agent (either human or software). This can be alleviated by employing the concept of view over LO metadata that enables personalization of the way that metadata is viewed by the agent. More precisely, creating a view over some data on the Semantic Web (a LO metadata in our case) essentially consists of the creation of a virtual metadata schema consistent with the agent's needs or perception of those data. The concept of view enables us to personalize the process of searching LORs and make it both easier and more efficient. To the best of our knowledge, currently there are only two Semantic Web view languages, both of them build upon RQL query language and aimed for RDF(S) data models: (1) The RDF View Language—RVL (Magkanaraki, Tannen, Christophides, & Plexousakis, 2003)—a view definition language capable of specifying not only views over RDF-encrypted metadata (i.e., virtual source descriptions) but also views over metadata schemas (i.e., virtual RDFS schemas); and (2) the view language proposed by Volz, Oberle, and Studer (2003). In addition, some query languages can be used for specifying views over data on the Semantic Web, as it was proposed in Miklos, Neumann, Zdun, and Sintek (2003).

5. Application Examples

In this section, we briefly sketch two educational Web applications aiming to further illustrate the proposed approach. The first one is intended for teaching Petri nets, while the second one targets the domain of the Intelligent Systems.

5.1 An Educational Web Application for Teaching Petri Nets

Figure 6 depicts the suggested approach as applied for teaching or learning Petri nets, a formal mathematical and graphical tool for system modelling, simulation, and analysis. Systems of this type have two kinds of users: teachers and students. A teacher creates LOs using P3 tool—a Petri net tool we have developed for teaching Petri nets (Gašević & Devedžić, 2004b). The P3 tool is based on the *Petri net ontology* (Gašević & Devedžić, 2004a). It is able to generate RDF description of a Petri net model as well as to produce an SVG-based graphical description of a Petri net model. In this context we considered a Petri net model in the RDF-annotated SVG form as a LO. The created LOs are incorporated in the Web application. Each Web page of the application contains a graphical presentation of an adequate Petri net model (RDF-annotated SVG) and provides support for simulation with that model. A user can save a Petri net that (s)he is working with in the PNML (Petri Net Markup Language) format, an XML-based standard for sharing Petri net models, and that Petri net can be further imported into Petri net tools (e.g., P3). The same model in the SVG format can be used in other Web pages, but also can be shown in a tool such as MS PowerPoint.

5.2 TANGRAM

TANGRAM is a Web application built on top of a LO repository and intended to be useful to both content authors and learners interested in the domain of Intelligent Systems (IS). It fully conforms to the aforementioned approach, and is devised to be an example of applying Semantic Web technologies, ontologies in particular, for achieving reuse and repurposing of the LOs' content. Furthermore, TANGRAM provides adaptation of learning content to the specific needs of individual learners. It leverages the explicitly defined structure of LOs to directly access each of the LOs'

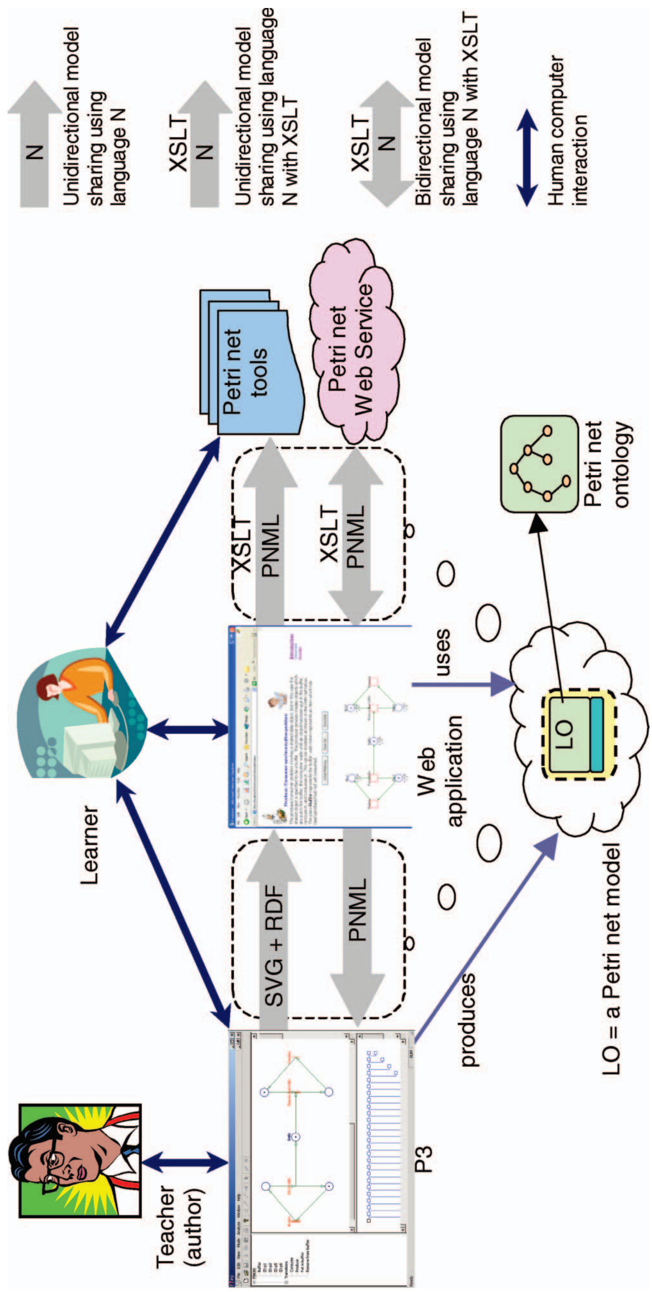

Figure 6. The educational Web application for teaching Petri nets that uses enhanced LOs

components and tailor them to the specific needs (i.e., knowledge background, preferences, and learning style) of individual learners. Besides being able to directly access components of a LO, TANGRAM is empowered to dynamically, on-the-fly create a new, personalized learning content out of existing components (Jovanović et al., 2006). Specifically, two basic functionalities of the system from the learners' perspective are:

- Provision of learning content adapted to the learner's current level of knowledge of the domain concept of interest, his or her learning style, and preferences regarding language and content author. However, TANGRAM does not aim to make a choice for the learner. Instead, the system provides guidance to the learner (using link annotation and hiding techniques—see section 3.3), and eventually lets him or her decide on the lesson or course to learn from.
- 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 IS).

To make the system capable of performing these functions and enable its seamless extension with other advanced services, we based its architecture on a number of ontologies. Besides the ALOCoM Content Structure ontology and the IS domain ontology, it makes use of (1) the ALOCoM Content Type ontology defining the pedagogical or instructional roles of LOs and their components (e.g., definition, example, exercise), (2) the Learning Paths ontology that formally represents an optimal learning path through domain topics, and (3) the User Model ontology that explicitly defines relevant information about TANGRAM's users (content authors and learners). These ontologies are available from http://iis.fon.bg.ac.yu/TANGRAM/home.html.

In TANGRAM a learning session starts after a registered learner selects a subdomain of IS to learn about (e.g., XML Technologies). Having verified the learner's knowledge of the chosen subdomain (using the IS domain ontology, the Learning Paths ontology, and the learner's ontology-based user model), TANGRAM builds a visual representation of that subdomain (i.e., its hierarchical organization of concepts) in the form of an annotated tree of links (the upper left corner of Figure 7), exploiting link annotation and link hiding techniques. Specifically, the following link annotations are used:

1. Blue bullet preceding a link to a domain concept denotes that the learner knows the topic that the link points to;
2. Green bullet denotes a recommended domain concept, that is, a concept that the learner has not learned yet, but has knowledge about all prerequisite topics;
3. Red bullet is used to annotate a domain topic that the learner is still not ready for as (s)he is ignorant of the prerequisite topics.

In addition, the link hiding technique is used to prevent the learner from accessing topics that are too advanced for him or her. In other words, links

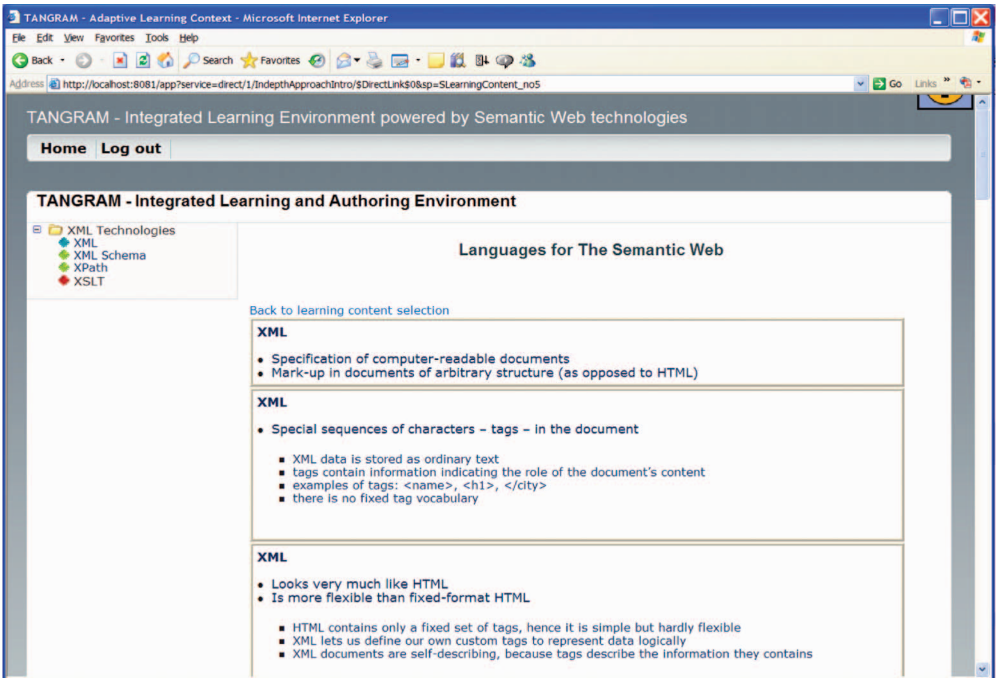

Figure 7. A screenshot of TANGRAM with an excerpt of dynamically built personalized learning content

annotated with red bullets are made inactive. Hence, the learner is free to choose one of the blue or green bulleted topics. As the selection is being made TANGRAM builds a personalized content on the selected topic, "bearing in mind" both the learner's learning style and his or her preferences regarding content authors. Specifically, TANGRAM initiates this process by querying the repository of LOs for content units (both LOs and their components) covering the selected domain topic. The query is based on the semantic annotations of the content units from the repository. If the repository does not contain content units on the selected topic, the system exploits the ontological relations between the domain concepts in order to determine the most suitable substitution for the targeted concept. Subsequently, the retrieved content units are grouped into lessons and sorted according to their inferred relevancy for the learner. Having finished this process, the system presents the learner with the descriptions of the generated lessons and allows him or her to choose the one to take. After the learner makes a selection, TANGRAM presents the lesson's content using its generic form for presentation of dynamically assembled learning content. Figure 7 presents TANGRAM's screenshot with an excerpt of a dynamically created, personalized content on the selected domain topic (XML in this case). Having presented the assembled learning content to the student, the system updates the learner model.

6. Related Work

In this section we try to identify some important research efforts covering relations between LOs and ontologies. So far, there have been many attempts to apply ontologies in the field of e-learning. A pioneer work on combining ontologies and metadata standards is presented in El Saddik, Fischer, and Steinmetz (2001). A more formal discussion about the use of ontologies in e-learning can be found in Lytras, Tsilira, and Themistocleous (2003). We have already mentioned the classification of ontologies in the e-learning domain given by Stojanović et al. (2001), as well as the classification proposed by Mohan and Brooks (2003). Apart from the classification, Mohan and Brooks also propose creating smarter LOs capable of performing many of the tasks typically associated with Learning Content Management Systems (e.g., to scan the Web looking for related LOs). We have an opposite attitude regarding this issue, that is, we propose making the semantics of the LO content more formal using ontologies. On top of such content intelligent learning services can be employed. Actually, the benefits stemming from the use of the Semantic Web technologies in the e-learning context have already been recognized by Panteleyev, Puzankoc, Sazykin, and Sergeyev (2002) as discovery of resources, composing new resources compliant to the requirements of a particular learner out of the available resources, and user-resource interaction dynamically adapted to the features of the particular user or device. In fact, the tools discussed in this paper are supposed to set the scene for those services. Furthermore, Dolog, Henze, Nejdl, and Sintek (2004) combine user model ontologies and e-learning ontology in order to achieve smart spaces for learning. Their approach is similar to the one upon which we based TANGRAM.

The use of the Semantic Web annotation tools for e-learning has already been discussed by Zarraonandía, Dodero, Diaz, and Sarasa (2004). The authors argue for the use of the IEEE LOM RDF binding (Nilsson et al., 2003), as well as extending the IEEE LOM schema and its classification category with additional markup elements. However, this approach does not assume annotating LO content. More closely related to our approach is the one which proposes adding semantic to media resources (Verhaart & Kinshuk, 2004).

An interesting approach to content reuse and adaptation in e-learning is given in Sommaruga (2004). This approach is oriented towards producing several presentation forms (e.g., slides, scientific report, etc.) of the same learning content. Each of the presentation forms is referred to as a LO type and for each of them the author proposes defining a generic syntactic form (e.g., abstract slide model). We believe that this approach has a limitation as it is based on XML schema with well-known incapacity for describing semantics (Klein, 2001). The second shortcoming is that this solution does not assume having one common LO type like the ALOCoM ontology has. The main benefit of this approach is that learning applications can reuse parts of the LO types they have not previously supported.

The approach to LO content repurposing proposed in this paper is similar to the model-driven solution for multimedia content repurposing (Obrenović, Starčević, & Selić, 2004). That solution is based on a common (i.e., unified) metamodel describing different aspects of multimedia, user interface modelling, as well as human-computer interaction in general. The common metamodel is an extensible platform, so that new concepts (and sets of concepts, i.e., profiles) can be introduced as its extensions. In our solution the ontology is an equivalent of the common metamodel.

7. Discussion

In a conventional LO life-cycle scenario, after being created, a LO is annotated with a standards-compliant set of metadata and stored in a LO repository. The annotations are supposed to facilitate search and retrieval of LOs relevant for a specific user or task (e.g., a teacher authoring a lesson). However, a conventional metadata-based search does not offer the required functionality. The example that follows illustrates this statement. In the example, we consider this paper as a LO that is to be stored in a LO repository. We present two scenarios, first a conventional one and subsequently an ontology-based one.

In a conventional scenario, our paper has to be annotated with LOM-compliant metadata before it can be uploaded to a LO repository. This means that one of us (i.e., the authors of the LO) is supposed to mark up the LO with (at least) the title, the date of publishing, the authors' names, the educational category, and so forth. How many of the numerous LOM elements will be filled out depends primarily on the time we can spend on the annotation work (which is not much) and our enthusiasm to make the annotations as detailed and precise as possible (although we are very enthusiastic about research, we become rather passive when administrative work appears). However, let us suppose that we were extremely industrious and we filled out all LOM metadata fields before storing our paper (i.e., LO) into the repository. Now, consider a teacher who is preparing a lesson on ontologies and ontology languages and who is searching the repository to find some suitable content for the lesson. The teacher issues a search using "ontology" as a keyword and optionally specifies some additional (metadata-based) search criteria. It is highly probable that the search engine will retrieve, among others, the LO from our example (i.e., this paper). It is not very probable that the teacher will be attracted by the title of the paper ("Ontology-based annotation of LO content" does not promise to say much about ontologies (and/or ontology languages) per se?), or the authors (a paper by Tim Berners Lee, Deborah McGuiness or Stephan Decker would look much more promising). However, the paper does have a couple of sections which provide a concise overview of ontologies, ontology libraries, and ontology query languages—content that can be readily included in a lesson on ontologies. Because of the lack of information in metadata and lack of time to go through every search result the teacher skips this paper—unfortunately, not only this one, but also many other relevant LOs.

The second scenario assumes structuring a LO in accordance with a content structure ontology and annotating both the LO and its components with concepts from the relevant domain ontology(-ies). This means that the LO from our example (i.e., this paper) is decomposed into sections, and each section into paragraphs, tables, and figures. Furthermore, each part is assigned a concept from the domain ontology, in this case, the ontology of the Semantic Web domain. As we already explained in the section on annotation tools (3.1.1), the technology is becoming increasingly available to make this process semi-automatic. Now, when the teacher searches the repository for learning content on ontologies, the search engine retrieves not only the full documents (LOs), but also content units of smaller granularity levels that exactly correspond to the teacher's information needs and are ready to be included in the lesson (s)he is authoring. In addition, enhanced with the ontology support the search engine is able to do approximate search. In other words, if it cannot find any LO that matches the submitted query, the search engine uses the domain ontology to reformulate the search request and provide the user (i.e., the teacher in this case) with potentially worthy search results. For example, if not able to find anything on "RQL" (an RDF query language), the search engine uses the domain ontology to find semantically close concepts: "RDQL" and "Corese" as kinds of RDF query languages are the closest matches, but also "OWL-QL" and "TRIPLE" as ontology query languages, although not based on RDF, might satisfy the user's needs. Similarly, the teacher may submit a query for an image of the "Semantic Web architecture" and be provided with appropriate figures taken out of different kinds of LOs discussing Semantic Web issues. Again this is possible with ontological structuring of LOs and semantic annotation of their content.

8. Conclusions

In this paper we presented an approach towards semantically enhancing LO content using ontologies. That way the notion of LOs reusability is extended—once created, LOs can be used not only in different courses, but also for different purposes and in different ways (e.g., presentation and platforms). Furthermore, by making LO content ontology-aware, we can improve retrieval of relevant LOs from LORs. To build LOs with semantically enhanced content we suggest an authoring environment that leverages domain ontologies, authoring and annotation tools and Semantic Web query languages. Furthermore, we propose an adaptive learning environment that makes use of LOs with semantically annotated content to provide personalized learning experience to its users (i.e., learners).

Currently, we are working on further extending the proposed ontological framework to enable capturing and formal representation of the context of use of LOs. We believe that the availability of explicitly defined context-related data will further facilitate discovery, reuse, and personalization of LOs. To describe a specific context of use of a LO, we need information about the course, module, or lesson that the LO was used in, the learning activity that the LO was used to support, the instructional or pedagogical role that the LO played in that activity, the learning objectives the LO was expected to fulfil, and the learner(s) who actually used the LO in that specific learning context. To enable explicit definition of these data, we have already developed an ontology for learning design, a user model ontology, and a competency ontology, and we are presently experimenting with the combined use of these ontologies to enable personalized views over LORs and recommendation of the most suitable LOs.

References

Azouaou, F., & Desmoulins, C. (in press). Teachers' document annotating: Models for a digital memory tool. *International Journal of Continuing Engineering Education and Lifelong Learning (IJCEELL)*, *16*.

Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D. L., Patel-Schneider, P. F., & Stein, L. A. (2004). OWL Web ontology language reference. W3C recommendation. Retrieved December 5, 2005, from http://www.w3.org/TR/2004/REC-owl-ref-20040210

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The Semantic Web. *Scientific American*, *284*(5), 34–43.

Bloom, B., & Krathwohl, D. (Eds.). (1956). *Taxonomy of educational objectives: The classification of educational goals: Handbook I, cognitive domain.* New York: Longman.

Brase, J., & Nejdl, W. (2004). Ontologies and metadata for elearning. In S. Staab & R. Studer (Eds.), *Handbook on ontologies* (pp. 555–574). Berlin: Springer-Verlag.

Brusilovsky, P. (2001). Adaptive hypermedia. *User Modelling and User-Adapted Interaction*, *11*(1–2), 87–110.

DAML Ontology Library. (2003). Retrieved April 5, 2006, from http://www.daml.org/ontologies/

Devedžić, V. (2004). Web intelligence and artificial intelligence in education. *Educational Technology & Society*, *7*(4), 29–39.

Dolog, P., Henze, N., Nejdl, W., & Sintek, M. (2004). Personalization in distributed elearning environments. In *Proceedings of the 13th International World Wide Web Conference on Alternate Track Papers & Posters* (pp. 170–179). New York, USA. Retrieved December 4, 2005 from http://www2004.org/proceedings/docs/contents.htm

Duval, E. (Ed.). (2002). 1484.12.1 IEEE standard for learning object metadata. Retrieved April 28, 2005, from http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf

Duval, E., & Hodgins, W. (2003). A LOM research agenda. In *Proceedings of the 12th International World Wide Web Conference* (pp. 1–9). Budapest, Hungary. Retrieved December 9, 2006, from http://www2003.org/cdrom/papers/alternate/P659/p659-duval.html.html

Dzbor, M., Motta, E., & Domingue, J. (2004). Opening up Magpie via semantic services. *The Semantic Web–ISWC 2004, Lecture Notes in Computer Science, 3298*, pp. 635–649. Berlin: Springer.

El Saddik, A., Fischer, S., & Steinmetz, R. (2001). Reusable multimedia content in web-based learning systems. *IEEE Multimedia*, *8*(3), 30–38.

Euzenat, J., Stuckenschmidt, H., & Yatskevich, M. (2005). Introduction to the ontology alignment evaluation 2005. In *Proceedings of the KCAP2005 Workshop on Integrating Ontologies* (pp. 61–71). Banff, Canada. Retrieved November 4, 2005, from http://oaei.ontologymatching.org/doc/intont2005proceedings.pdf

Fikes, R., Hayes, P., & Horrocks, I. (2003). *OWL-QL—A language for deductive query answering on the Semantic Web* (KSL Tech. Rep. 03-14). Stanford, CA: KSL, Stanford University.

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). *Design patterns: Elements of reusable object-oriented software.* Reading, MA: Addison-Wesley.

Gašević, D., & Devedžić, V. (2004a). Reusing Petri nets through the Semantic Web. *The Semantic Web: Research and applications. Lecture Notes in Computer Science, 3053*, pp. 284–298. Berlin: Springer.

Gašević, D., & Devedžić, V. (2004b). Teaching Petri nets using P3. *Educational Technology & Society*, *7*(4), 153–166.

Gašević, D., Jovanović, J., & Devedžić, V. (2004). Enhancing learning object content on the Semantic Web. In *ICALT* (pp. 714–716). Fourth IEEE International Conference on Advanced Learning Technologies (ICALT'04).

Gruber, T. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, *5*(2), 199–220.

Handschuh, S., & Staab, S. (2002). Authoring and annotation of Web pages in CREAM. In *Proceedings of the 11th International World Wide Web Conference* (pp. 462–473). Honolulu, USA. Retrieved November 5, 2005, from http://www2002.org/CDROM/refereed/506/index.html

Handschuh, S., Volz, R., & Staab, S. (2003). Annotation for the deep Web. *IEEE Intelligent Systems*, *18*(5), 42–48.

Hendler, J. (2001). Agents and the Semantic Web. *IEEE Intelligent Systems*, *16*(2), 30–37.

IMS Learning Design Information Model. (2003). Version 1.0 Final Specification, rev.20, 2003. Retrieved January 18, 2006, from http://www.imsglobal.org/learningdesign/ldv1p0/imsld_infov1p0.html

Jovanović, J., Gašević, D., & Devedžić, V. (2005). TANGRAM: An ontology-based learning environment for intelligent information systems. In G. Richards (Ed.), *Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education* (pp. 2966–2971). Chesapeake, VA: AACE.

Jovanović, J., Gašević, D., & Devedžić, V. (2006). Dynamic assembly of personalized learning content on the Semantic Web. *The Semantic Web: Research and applications. Lecture Notes in Computer Science, 4011*, pp. 544–558. Berlin: Springer.

Kalyanpur, A., Hendler, J., Parsia, B., & Golbeck, J. (2003). SMORE—Semantic markup, ontology, and RDF editor. Retrieved April 28, 2005, from http://www.mindswap.org/papers/SMORE.pdf

KAON Semantic Web tool suite. 2005. Retrieved April 20, 2006, from http://kaon.semanticweb.org/

Karvounarakis, G., Alexaki, S., Christophides, V., Plexousakis, D., & Scholl, M. (2002). RQL: A declarative query language for RDF. In *Proceedings of the 11th International World Wide Web Conference* (pp. 592–603). Honolulu, USA. Retrieved November 1, 2005, from http://www.2002.org/CDROM/refereed/329/index.html

Keenoy, K., Levene, M., & Peterson, D. (2003). Personalisation and trails in self e-learning networks, WP4 deliverable 4.2. Retrieved April 28, 2005, from http://www.dcs.bbk.ac.uk/selene/reports/Del4.2-2.1.pdf

Keenoy, K., Poulovassilis, A., Christophides, V., Rigaux, P., Papamarkos, G., Magkanaraki, A., Stratakis, M., Spyratos, N., & Wood, P. (2004). Personalisation services for self e-learning networks. Retrieved April 28, 2005, from http://www.dcs.bbk.ac.uk/selene/reports/SeLeNe-Personalisation.pdf

Klein, M. (2001). XML, RDF, and relatives. *IEEE Intelligent Systems*, *16*(2), 26–28.

Lanfranchi, V., Ciravegna, F., & Petrelli, D. (2005). Semantic Web-based document: Editing and browsing in AktiveDoc. *The Semantic Web: Research and applications. Lecture Notes in Computer Science, 3532*, pp. 623–632. Berlin: Springer.

Lytras, M., Tsilira, A., & Themistocleous, M. (2003). Towards the semantic e-learning: An ontological oriented discussion of the new research agenda in e-learning. In *Proceedings of the 9th Americas Conference on Information Systems* (pp. 2985–2997). Berlin: Springer Verlag.

Maedche, A., & Staab, S. (2001). Ontology learning for the Semantic Web. *IEEE Intelligent Systems, Special Issue on Semantic Web*, *16*(2), 72–79.

Magkanaraki, A., Tannen, V., Christophides, V., & Plexousakis, D. (2003). Viewing the semantic web through RVL lenses. *The Semantic Web–ISWC 2002. Lecture Notes in Computer Science, 2342*, pp. 96–112. Berlin: Springer.

Manola, F., & Miller, E. (Eds.). (2003). Resource description framework (RDF) primer. W3C proposed recommendation. Retrieved December 7, 2005, from http://www.w3.org/TR/2003/PR-rdf-primer-20031215/

McClelland, M. (2003). Metadata standards for educational resources. *IEEE Computer*, *36*(11), 107–109.

Miklos, Z., Neumann, G., Zdun, U., & Sintek, M. (2003). Querying semantic web resources using triple views. *The Semantic Web–ISWC 2003. Lecture Notes in Computer Science*, *2870*, pp. 517–532. Berlin: Springer.

Mohan, P., & Brooks, C. (2003). Learning objects on the Semantic Web. In *Proceedings of the 3rd IEEE International Conference on Advanced Learning Technologies* (pp. 195–199). Los Alamitos, CA: IEEE Computer Society Press.

Nilsson, M., Palmer, M., & Brase, J. (2003). The LOM RDF binding—Principles and implementation. In *Proceedings of the 3rd Annual ARIADNE Conference*. Leuven, Belgium.

Noy, N. F. (2004). Semantic integration: A survey of ontology-based approaches. *SIGMOD Record*, *33*(4), 65–70.

Obrenović, Ž., Starčević, D., & Selić, B. (2004). A model driven approach to content repurposing. *IEEE Multimedia*, *11*(1), 62–71.

OwlSeek. (2005). Retrieved May 25, 2006, from http://www.owlseek.com/

Panteleyev, M., Puzankov, D., Sazykin, P., & Sergeyev, D. (2002). Intelligent educational environments based on the Semantic Web technologies. In *Proceedings of the IEEE International Conference on Artificial Intelligence Systems* (pp. 457–462). Los Alamitos, CA: IEEE Computer Society Press.

Paquette, G. (2003). *Instructional Engineering for Network-Based Learning*. San Francisco: Pfeiffer/Wiley.

Priestley, M. (2001). DITA XML: A reuse by reference architecture for technical documentation. In *Proceedings of the 19th International Conference on Computer Documentation* (pp. 152–156). New York: ACM Press.

SchemaWeb. (2004). Retrieved May 15, 2006, from http://www.schemaweb.info/

Seaborne, A. (2004). RDQL—A query language for RDF, W3C member submission. Retrieved April 28, 2005, from http://www.w3.org/Submission/RDQL/

Shvaiko, P., & Euzenat, J. (2005). A survey of schema-based matching approaches. *Journal on Data Semantics IV*, *LNCS 3730*, 146–171.

Sintek, M., & Decker, S. (2002). TRIPLE—A query, inference, and transformation language for the Semantic Web. *The Semantic Web–ISWC 2002. Lecture Notes in Computer Science*, *2342*, pp. 364–378. Berlin: Springer.

Sommaruga, L. (2004). An approach to content re-usability and adaptability in e-learning. In P. Kommers & G. Richards (Eds.), *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications* (pp. 2098–2104). Chesapeake, VA: AACE.

Stojanović, L., Staab, S., & Studer, R. (2001). eLearning in the Semantic Web. In *Proceedings of the World Conference on the WWW and the Internet (WebNet 2001)* (pp. 1174–1183). Chesapeake, VA: AACE.

Swoogle. (2005). Retrieved April 6, 2006, from http://swoogle.umbc.edu/

Tallis, M. (2003). Semantic word processing for content authors. In *Proceedings of the workshop notes of the Knowledge Markup and Semantic Annotation Workshop*. Sanibel, USA.

TANGRAM. (2006). Retrieved March 18, 2006, from http://iis.fon.bg.ac.yu/TANGRAM/home.html

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: Science, Services and Agents on the World Wide Web, 4*(1), 14–28.

Verbert, K., Gašević, D., Jovanović, J., & Duval, E. (2005). Ontology-based learning content repurposing. In *Proceedings of the 14th International World Wide Web Conference* (pp. 1140–1141). Chiba, Japan. New York: ACM Press.

Verbert, K., Klerkx, J., Meire, M., Najjar, J., & Duval, E. (2004). Towards a global component architecture for learning objects: An ontology based approach. *Lecture Notes in Computer Science, 3292*, pp. 713–722. Berlin: Springer.

Verhaart, M., & Kinshuk (2004). Adding semantics and context to media resources to enable efficient construction of learning objects. In *Proceedings of the 4th IEEE International Conference on Advanced Learning Technologies* (pp. 651 – 653). Los Alamitos, CA: IEEE Computer Society Press.

Volz, R., Oberle, D., & Studer, R. (2003). Views for light-weight web ontologies. In *Proceedings of the ACM Symposium on Applied Computing* (pp. 1168 – 1173). New York: ACM Press.

Zarraonandía, T., Dodero, J. M., Díaz, P., & Sarasa, A. (2004). Domain ontologies integration into the learning objects annotation process. In *Proceedings of the 2nd International Workshop on Applications of Semantic Web Technologies for E-Learning (at ITS2004)*. Retrieved December 10, 2005, from http://www.win.tue.nl/SW-EL/2004/swel-its-program.html

Appendix

Ontology was initially defined as a formal specification of a conceptualization (Gruber, 1993). A more recent definition states that an ontology is a set of knowledge terms, including the vocabulary, the semantic interconnections, and some simple rules of inference and logic for some particular topic (Hendler, 2001).

Resource Description Framework (RDF) is a World Wide Web Consortium (W3C) standard language for describing resources on the Web, that is, for specifying metadata for Web resources (Manola & Miller, 2003). RDF is based on the idea that things (resources) being described have properties which have values, and that resources can be described by making statements that specify those properties and values.

Ontology Web Language (OWL) is a semantic markup language for publishing and sharing ontologies on the Web (Bechhofer et al., 2004). It is developed as a vocabulary extension of RDF. Last year OWL become the W3C standard ontology language.

Intelligent Agents (Autonomous Agents) are autonomous software entities that acquire information from their environment (“perceive” the environment), decide about their actions, and perform them. They can help users in different ways: hide the complexity of difficult tasks; perform some tasks on behalf of their users; teach the end users; monitor events and procedures; help the users collaborate and cooperate, and so forth.