International Journal of Artificial Intelligence in Education 14 (2004) 39-65
IOS Press

Education and the Semantic Web

Vladan Devedzic, *Department of Information Systems and Technologies, FON – School of Business Administration, University of Belgrade, POB 52, Jove Ilica 154, 11000 Belgrade, Serbia and Montenegro*
devedzic@galeb.etf.bg.ac.yu
http://galeb.etf.bg.ac.yu/~devedzic/

Abstract. Recent developments in Web technologies and using AI techniques to support efforts in making the Web more intelligent and provide higher-level services to its users have opened the door to building the Semantic Web. That fact has a number of important implications for Web-based education, since Web-based education has become a very important branch of educational technology. Classroom independence and platform independence of Web-based education, availability of authoring tools for developing Web-based courseware, cheap and efficient storage and distribution of course materials, hyperlinks to suggested readings, digital libraries, and other sources of references relevant for the course are but a few of a number of clear advantages of Web-based education. However, there are several challenges in improving Web-based education, such as providing for more adaptivity and intelligence. Developments in the Semantic Web, while contributing to the solution to these problems, also raise new issues that must be considered if we are to progress. This paper surveys the basics of the Semantic Web and discusses its importance in future Web-based educational applications.

Instead of trying to rebuild some aspects of a human brain, we are going to build a brain of and for humankind.
D. Fensel and M.A. Musen (Fensel & Musen, 2001)

INTRODUCTION

One of the hottest R&D topics in recent years in the AI community, as well as in the Internet community, is *the Semantic Web*. It is about making the Web more understandable by machines (Heflin & Hendler, 2001). It is also about building an appropriate infrastructure for intelligent agents to run around the Web performing complex actions for their users (Hendler, 2001). In order to do that, agents must retrieve and manipulate pertinent information, which requires seamless agent integration with the Web and taking full advantage of the existing infrastructure (such as message sending, security, authentication, directory services, and application service frameworks) (Scott Cost et al., 2002). Furthermore, Semantic Web is about explicitly declaring the knowledge embedded in many Web-based applications, integrating information in an intelligent way, providing semantic-based access to the Internet, and extracting information from texts (Gómez-Pérez & Corcho, 2002). Ultimately, Semantic Web is about how to implement reliable, large-scale interoperation of Web services, to make such services computer interpretable

1560-4292/03/$8.00

- to create a Web of machine-understandable and interoperable services that intelligent agents can discover, execute, and compose automatically (McIlraith, Son, & Zeng, 2001).

Why do we need all that? Isn't the Web an immense, practically unlimited source of information and knowledge that everyone can use?

The problem is that the Web is huge, but not smart enough to easily integrate all of those numerous pieces of information from the Web that a user really needs. Such integration at a high, user-oriented level is desirable in nearly all uses of the Web. Today, most Web information is represented in natural-language; however, our computers cannot understand and interpret its meaning. Humans themselves can process only a tiny fraction of information available on the Web, and would benefit enormously if they could turn to machines for help in processing and analyzing the Web contents (Noy et al., 2001). Unfortunately, the Web was built for human consumption, not for machine consumption - although everything on the Web is *machine-readable*, it is not *machine-understandable* (Lassila, 1998). We need the Semantic Web to express information in a precise, machine-interpretable form, ready for software agents to process, share, and reuse it, as well as to understand what the terms describing the data mean. That would enable Web-based applications to interoperate both on the syntactic and semantic level.

Note that it is Tim Berners-Lee himself that pushes the idea of the Semantic Web forward. The father of the Web first envisioned a Semantic Web that provides automated information access based on machine-processable semantics of data and heuristics that use these metadata (Berners-Lee, Hendler, & Lassila, 2001; Berners-Lee, Fischetti, & Dertouzos, 1999). The explicit representation of the semantics of data, accompanied with domain theories (that is, ontologies), will enable a Web that provides a qualitatively new level of service - for example, intelligent search engines, information brokers, and information filters (Decker et al., 2000; Fensel & Musen, 2001).

People from the World Wide Web Consortium (W3C) already develop new technologies for web-friendly data description (see www.w3.org/XML and www.w3.org/RDF). Moreover, AI people have already developed some useful applications and tools for the Semantic Web (Noy et al., 2001; Scott Cost et al., 2002).

In education, we should pay close attention to such developments and trends. This paper surveys important issues related to the development of the Semantic Web, and then discusses their implications for Web-based teaching and learning. It describes what it means precisely to create, to find, and to use educational resources on the Semantic Web pages, as opposed to doing it on today's Web. It also suggests a way from just a vision of putting machine-understandable educational material on the Web to making machines really using and interpreting it automatically when numerous educators, authors and learners interact with the Web. The proposed way requires familiarization with new technologies first, as a firm and stable foundation for developing next-generation Web-based intelligent educational software. It also stresses the need for making future AIED systems better engineered than current ones. The paper presents the background and context for activities of developing Semantic Web-based educational systems, describes the current state of the development, indicates some existing applications and tools, and introduces some future possibilities that might emerge when machines can read the Semantic Web.

SEMANTIC WEB

There is a number of important issues related to the Semantic Web. Roughly speaking, they belong to four categories: languages for the Semantic Web, ontologies, semantic markup of pages on the Semantic Web, and services that the Semantic Web is supposed to provide.

Languages for the Semantic Web

In order to represent information on the Semantic Web and simultaneously make that information both syntactically and semantically interoperable across applications, it is necessary to use specific languages. It is important for Semantic Web developers to agree on the data's syntax and semantics before hard-coding them into their applications, since changes to syntax and semantics necessitate expensive application modifications (Wuwongse, Anutariya, Akama, & Nantajeewarawat, 2002).

There are a lot of such languages around. Some of them are higher-level ones (discussed in the next subsection), others are lower-level. One way or another, most of them are based on *XML* (eXtensible Markup Language), *XML Schemas*, *RDF* (Resource Definition Framework), and *RDF Schemas*, all four developed under the auspices of W3C and using XML syntax (Klein, 2001).

Figure 1 shows an example of representing the same piece of information in HTML and in XML. While HTML is layout-oriented, XML is more structure-oriented. HTML is based on a fixed set of tags to format text; in XML, tags are arbitrary (user-defined) and bear some semantic information themselves.

XML Schema provides the necessary framework for creating XML documents by specifying the valid structure, constraints, the number of occurrences of specific elements, default values, and data types to be used in the corresponding XML documents, Figure 2. The encoding syntax of XML Schema is XML, and just like XML itself XML Schema documents use *namespaces* that are declared using the *xmlns* attribute. Namespaces define contexts within which the corresponding tags and names apply.

```
<UL>
<LI>E. Wenger, <EM>Artificial intelligence and tutoring systems:
Computational approaches to the communication of knowledge</EM>,
Morgan/Kaufmann Publishing Co., Los Altos, CA, 1987.
</UL>
```

(a)

```
<BOOK>
    <AUTHOR> E. Wenger </AUTHOR>
    <TITLE> Artificial intelligence and tutoring systems:
Computational approaches to the communication of knowledge </TITLE>
    <PUBLISHER> Morgan/Kaufmann Publishing Co. </PUBLISHER>
    <YEAR> 1987 </YEAR>
</BOOK>
```

(b)

Fig. 1. a) A piece of HTML code b) The same information in XML code

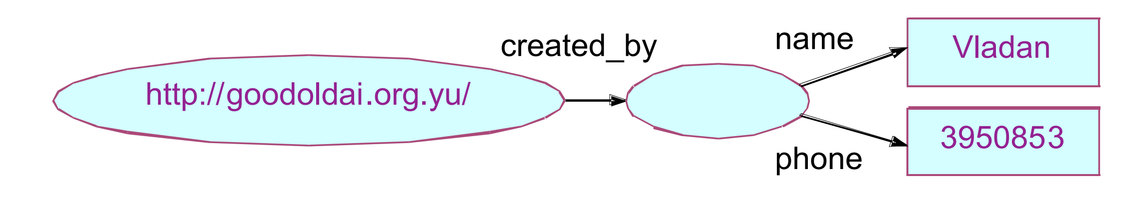
```
<xsd:schema xmlns:xsd="http://www.w3.org/1999/XMLSchema">
<xsd:element name="BOOK" type="BOOKTYPE"/>
<xsd:complexType name="BOOK_TYPE" >
    <xsd:element name="AUTHOR" type="xsd:string"
        minOccurs="1" maxOccurs="unbounded"/>
    <xsd:element name="TITLE" type="xsd:string"/>
    . . .
    <xsd:attribute name="isbn" type="xsd:string"/>
</xsd:complexType>
</xsd:schema>
```

Fig. 2. An example of XML Schema

RDF is a framework to represent data about data (metadata), and a model for representing data about "things on the Web" (resources). One way to do it is to use O-A-V triplets or statements, as in Figure 3a. Each statement is essentially a relation between an object (a resource), an attribute (a property), a value (a resource or free text). Alternatively, each RDF model can be represented as a directed labelled graph, Figure 3b, or in an XML-based encoding.

OBJECT	ATTRIBUTE	VALUE
http://goodoldai.org.yu/	created_by	#anonymous_resource1
#anonymous_resource1	name	"Vladan"
#anonymous_resource1	phone	"3950853"

(a)



(b)

Fig. 3. a) A simple RDF model as a set of O-A-V triplets b) the equivalent directed labelled graph

Regardless of the representation syntax, RDF models use traditional knowledge representation techniques order to provide better semantic interoperability (traditionally, O-A-V triplets are natural semantic units for representing a domain). Still, an RDF model just provides a domain-neutral mechanism to describe metadata, but does not define (a priori) the semantics of any application domain.

RDF Schema (RDFS) defines the vocabulary of an RDF model. It provides a mechanism to define domain-specific properties and classes of resources to which those properties can be applied, using a set of basic modelling primitives (*class*, *subclass-of*, *property*, *subproperty-of*, *domain*, *range, type*). An RDFS can be specified using RDF encoding. Figure 4 shows an example. However, RDFS is rather simple and it still doesn't provide exact semantics of a domain.

```
<rdfs:Class rdf:ID="book">
    <rdfs:subClassOf rdf:resource="#publication"/>
    <rdfs:subClassOf>
    ...
    </rdfs:subClassOf>
</rdfs:Class>
```

Fig. 4. An example of RDF Schema code

Ontologies

An ontology comprises 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). Ontologies applied to the Web are creating the Semantic Web (Fensel, van Harmelen, Horrocks, McGuinness & Patel-Schneider, 2001). Ontologies provide the necessary armature around which knowledge bases should be built (Swartout & Tate, 1999), and set grounds for developing reusable Web-contents, Web-services, and applications (Devedzic, 2001). Ontologies facilitate knowledge sharing and reuse, i.e. a common understanding of various contents that reaches across people and applications.

Technically, an ontology is a text-based piece of reference-knowledge, put somewhere on the Web for agents to consult it when necessary, and represented using the syntax of an *ontology-representation language*. There are several such languages around for representing ontologies (see (Gómez-Pérez & Corcho, 2002) for an overview and comparison of them). It is important to understand that most of them are built on top of XML and RDF. Figure 5 shows a piece of a simple ontology developed using the ontology-representation language called *OIL* (Ontology Inference Layer) (Horrocks et al., 2002). The equivalent RDFS representation uses the *oil* namespace to refer to the language primitives not supported by RDFS in its original form.

By 2004, the most popular higher-level ontology-representation languages were OIL and *DAML+OIL* (Horrocks & van Harmelen, 2002). An ontology developed in any such language is usually converted into an RDF/XML-like form and can be partially parsed even by common RDF/XML parsers (see www.w3.org/XML and www.w3.org/RDF for more information on such parsers). Of course, language-specific parsers are necessary for full-scale parsing. There is a methodology for converting an ontology developed in a higher-level language into RDF or RDFS (Decker et al., 2000).

In early 2004, W3C has officially released *OWL* (Web Ontology Language) as W3C Recommendation for representing ontologies (http://www.w3.org/TR/2004/REC-owl-features-20040210/). OWL is developed starting from description logic and DAML+OIL. The increasing popularity of OWL might lead to its widest adoption as the standard ontology representation language on the Semantic Web in the future. Essentially, OWL is a set of XML elements and attributes, with well-defined meaning, that are used to define terms and their relationships (e.g., *Class*, *equivalentProperty*, *intersectionOf*, *unionOf*, etc.). OWL elements extend the set of RDF and RDFS elements, and the *owl* namespace is used to denote OWL encoding.

In practice, ontologies are often developed using integrated, graphical, *ontology-authoring tools*, such as Protégé-2000 (http://protege.stanford.edu/), OILed (http: //img.cs.man.ac.uk/oil), and OntoEdit (http://ontoserver.aifb.uni-karlsruhe.de/ontoedit). They are used to develop new ontologies and modify existing ones. They let the author edit and develop ontologies concentrating on the domain's concepts and relationships, without worrying much about ontology-representation languages. The author can choose ontologies from a list, choose attributes and relations from another list, edit, add, remove, and merge ontologies. The output is usually produced in a specific high-level ontology-representation language such as OWL, in RDF/RDFS, in HTML, or in plain text.

```
class-def defined herbivore
    subclass-of animal, NOT carnivore
    slot-constraint eats
        value-type plant
        OR (slot-constraint is-part-of has-value plant)
```

(a)

```
<rdfs:Class rdf:ID="herbivore">
    <rdf:type rdf:resource="http://www.ontoknowledge.org/oil/RDFS-
schema/#DefinedClass"/>
    <rdfs:subClassOf rdf:resource="#animal"/>
    <rdfs:subClassOf>
        <oil:NOT>
            <oil:hasOperand rdf:resource="#carnivore"/>
        </oil:NOT>
    </rdfs:subClassOf>
</rdfs:Class>
```

(b)

Fig. 5. a) A simple ontology defined in OIL b) an equivalent ontology in RDFS (after (Fensel et al., 2001))

Services

Intelligent, high-level services like information brokers, search agents, information filters, intelligent information integration, and knowledge management, are what the users want from the Semantic Web. They are possible only if a number of ontologies populate the Web, enabling semantic interoperation between the agents and the applications on the Semantic Web, i.e. semantic mappings between terms within the data, which requires content analysis.

One specific kind of ontology is necessary to enable high-level Semantic Web services - ontologies of services themselves (McIlraith et al., 2001; Preece & Decker, 2002). These ontologies should include a machine-readable description of services (as to how they run), the consequences of using the service (e.g., the fee), and an explicit representation of the service logic (e.g., automatic invocation of another service). Services have their properties, capabilities, interfaces, and effects, all of which must be encoded in an unambiguous, machine-understandable form, to enable agents to recognize the services and invoke them automatically.

For example, an agent coming to a digital library to retrieve a specific bibliographical item on behalf of its user must be able to determine:

- how to find the library's Web page;
- how to invoke the search facility;
- what arguments to pass;
- what kind of results to expect (e.g., just the abstract or the full text, the text formats available);
- what are the conditions of retrieving the reference (e.g., cost, subscription, special offer).

The agent will then reason about these issues and, provided that there are no collisions with its internal logic, will automatically invoke the service eventually. Note that this is completely different from the current situation, in which the user must first discover the digital library manually, using a search engine, then read the discovered Web page, and also fill in the forms of the service manually.

Semantic markup

Ontologies merely serve to standardize and provide interpretations for Web content, but are not enough to build the Semantic Web. To make Web content machine-understandable, Web pages and documents themselves must contain semantic markup, i.e. annotations which use the terminology that one or more ontologies define and contain pointers to the network of ontologies, Figure 6. Semantic markup persists with the document or the page published on the Web, and is saved as part of the file representing the document/page. Services also must be properly marked-up, to make them computer-interpretable, use-apparent, and agent-ready. They must contain pointers to the corresponding service ontologies.

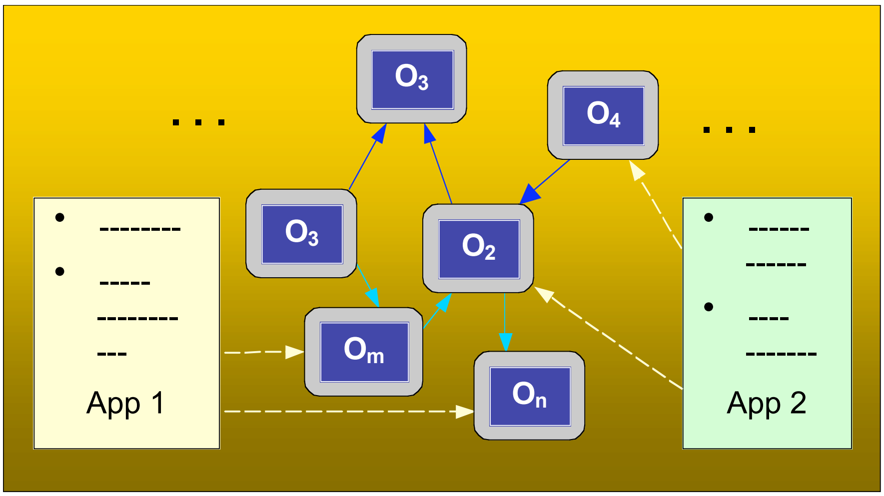


Fig. 6. Semantic markup provides mappings between Web pages and ontologies (O_i - ontologies)

Semantic markup of a Web page, document, or service might state that a particular entity is a member of a class, an entity has a particular property, two entities have some relationship between them, and that descriptions from different people refer to the same entity. Typically, semantic markup is published using an XML encoding for a high-level ontology-representation language syntax (Hendler & Heflin, 2001; Tallis, Goldman, & Balzer, 2002).

Using ontologies as references in marking-up pages and services on the Semantic Web enables knowledge-based indexing and retrieval of services by intelligent agents, agent brokers and humans alike, as well as automated reasoning about the services, such as how to use them, what parameters to supply, and what results to expect.

The annotation is done by using appropriate tools. These tools can be part-of or integrated with ontology-authoring tools, such as OIL tools (Fensel et al., 2001). They can also be standalone tools, such as the Knowledge Annotator tool (Hendler & Heflin, 2001). Furthermore, they can operate through a COTS tool, as in the case of the Briefing Associate tool that uses MS PowerPoint GUI (Tallis et al., 2002). Finally, they can be integrated with specific Semantic Web applications. An example of this last approach is ITtalks, a fielded application that facilitates user and agent interaction for locating talks on information technology (Scott Cost et al., 2002), which automatically generates DAML+OIL descriptions (markup) of user profiles when they register.

In all these approaches, authors need not necessarily understand the details of the markup process. They merely set the stage for the automatic markup process performed by the tool itself, by specifying the semantic context of the document through making selections of closely-related ontologies and filling in forms. The tools are ontology-aware, i.e. they offer the author a list of suitable ontologies to choose from and root the document in. Authoring tools with semantic markup authoring capabilities make the semantic markup a regular activity, without putting additional burden on the user. This way, the markup can be the product of many individual authors working independently. It can evolve over time along with the document, to accommodate changes in vocabularies, resolve conflicts, and scale up or down.

IMPLICATIONS FOR EDUCATION

What we're seeing is just the first version of the Web.
D. Fensel and M.A. Musen (Fensel and Musen, 2001)

Thousands of Web-based courses and other educational applications have been made available on the Web in recent years - see (Brusilovsky, 1999) and (Brusilovsky & Miller, 2001) for good surveys. *Intelligent Web-based educational* systems, as a kind of such applications, have been around for several years already. They are specific in that they introduce some amount of intelligence and adaptivity in Web-based teaching and learning. Intelligence of a Web-based educational system means the capability of demonstrating some form of knowledge-based reasoning in curriculum sequencing, in analysis of the student's solutions, and in providing interactive problem-solving support (possibly example-based) to the student, all adapted to the Web technology (Brusilovsky & Miller, 2001). Adaptivity can take different forms, such as (Brusilovsky, 1999):

- collecting some data about the student working with the system and creating the student model;

- adapting the presentation of the course material, navigation through it, its sequencing, and its annotation, to the student;
- using models of different students to form a matching group of students for different kinds of collaboration;
- identifying the students who have learning records essentially different from those of their peers (e.g., the students progressing too slow or too fast) and acting accordingly (e.g., show additional explanations, or present more advanced material).

There has been considerable success in building and using intelligent and adaptive Web-based educational applications. However, much more can and should be done. In the context of the Semantic Web, intelligent Web-based education takes on new dimensions.

The setting

Teaching, learning, collaboration, assessment, and other educational activities on the Semantic Web happen in the setting depicted in Figure 7. Intelligent *pedagogical agents* provide the necessary infrastructure for knowledge and information flow between the clients and the servers. They are autonomous software entities that support human learning by interacting with students/learners and authors/teachers and by collaborating with other similar agents, in the context of interactive learning environments (Johnson, Rickel, & Lester, 2000). Pedagogical agents help very much in locating, browsing, selecting, arranging, integrating, and otherwise using educational material from different *educational servers*. Pedagogical agents can support both collaborative and individualized learning, as well as the students' cognitive processes.

Pedagogical agents access *educational content* on a server by using high-level *educational services* shown in Figure 8, and the server possesses enough intelligence to arrange for *personalization* of the learning tasks it supports. In fact, from the learner's perspective the server appears to act as an intelligent tutor with both *domain* and *pedagogical* knowledge to conduct a learning session. It uses a *presentation planner* to select, prepare, and adapt the domain material to show to the student. It also gradually builds the *student model* during the session, in order to keep track of the student's actions and learning progress, detect and correct his/her errors and misconceptions, and possibly redirect the session accordingly.

Authors develop educational content on the server in accordance with important pedagogical issues such as instructional design and human learning theories, to ensure educational justification of learning, assessment, and possible collaboration among the students. The way to make the content machine-understandable, machine-processable, and hence agent-ready, is to provide semantic markup with pointers to a number of shareable *educational ontologies*. For developing educational ontologies, higher-level ontology-representation languages (languages built on top of XML/RDF) are currently a good choice. It is up to the developers of authoring tools to provide support for creating Web pages with educational content that points to appropriate ontologies and with educational services that ensure easy and automatic access of the content by means of pedagogical agents.

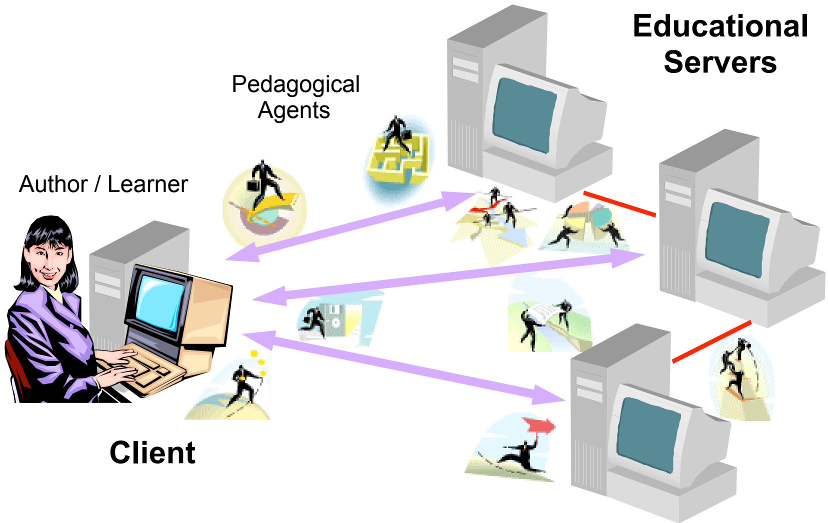


Fig. 7. The setting for Semantic Web-based education

Common prerequisites

If the above setting for education on the Semantic Web is to be established properly, then each development project should take care of available technological support and current technological trends. In practice, this means that the project should not start somewhere out of the mainstream of actual WWW technology trends if we want a good chance for the final outcome - a truly intelligent Web-based educational application - to be actually used by the students and the teachers and hence become really useful. Precisely, the project should exploit state-of-the-art standards, languages, and tools support provided by W3C and fit into the scheme popularly called *the Semantic Web "layer cake"* (Berners-Lee et al., 1999; Hendler, 2001), Figure 9.

Developers of authoring tools must provide means for creating educational Web pages and contents with ontological information. Most users of such tools (the authors) should not be expected to be experts in ontological engineering. On the contrary, authoring tools must let them insert ontological annotations in the documents they create transparently, through normal computer use. The minority of authors, of course, *will* have to develop suitable domain ontologies and pedagogical ontologies first (Mizoguchi & Bourdeau, 2000) (see the next subsection). Still, most other users need not even know that ontologies exist, and will still do free markup (Hendler, 2001). To provide that, one suitable approach that tool developers can take is to mark the contents from the libraries that come with the tools with pointers to ontologies. For example, the author of an intelligent Web-based tutor that teaches geometry may want to insert a drawing of a square into a certain document that learners may subsequently want to see. If the drawing has associated pointers to the ontologies of edges and vertices, saving the document as a HTML page will automatically create a markup for a pedagogical agent to understand the context of the document.

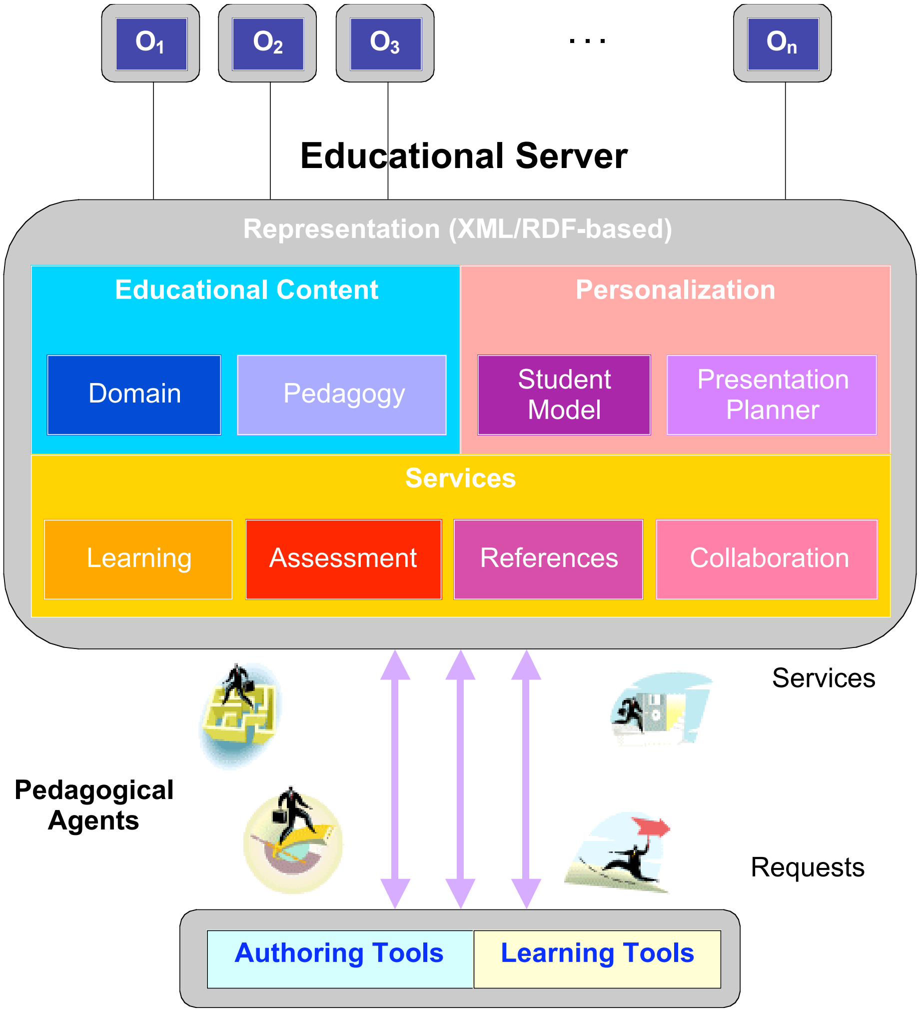


Fig. 8. Inside an educational server (O_i - ontologies)

Objectives and effects of providing semantic markup of educational material on the Web are two-fold:

- Using an interactive learning environment, the learners can query the Semantic Web for educational material by first choosing the relevant ontology (or ontologies); that establishes the context for the query.
- Pedagogical agents can crawl Web pages searching for markup and come up with relevant material. They can also collaborate with other pedagogical agents that will match the material found with the learner's knowledge level and preferences (as to what presentation format to use, or what teaching strategy to employ). The point is that the learner does not need to perform the discovery of the relevant educational contents manually.

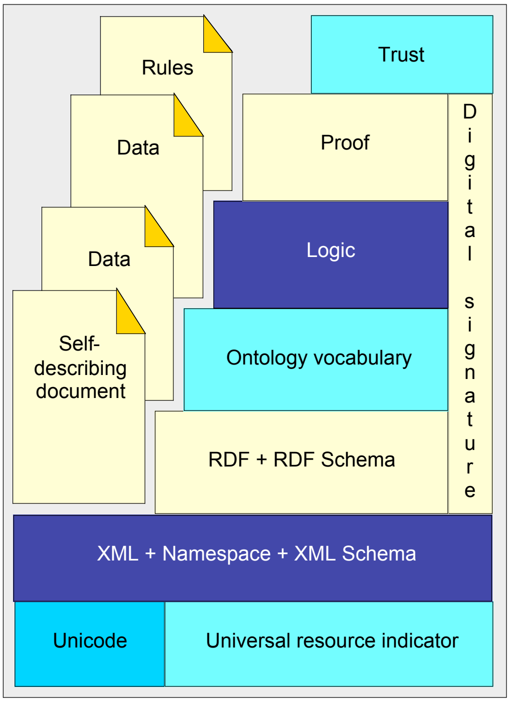


Fig. 9. Semantic Web "layer cake" (after (Berners-Lee et al., 1999))

What exactly needs to be marked up in order to enable pedagogical agents to automatically search, locate, retrieve, filter, and present educational material to the user? Possible answers include the following:

- educational services themselves; for example, services for retrieving "further-readings" material from digital libraries (these would roughly correspond to the services labelled "References" in Figure 8);
- user and group constraints and preferences, such as interests in specific course levels;
- agent procedures, such as an assessment procedure (procedures are (partial) compositions of existing educational Web services; they are designed to perform a particular task and marked-up for sharing and reuse by other users).

How to mark up material on educational servers to make it pedagogical agent-ready? Following the general ideas about markup on the Semantic Web suggested by McIlraith et al. (McIlraith et al., 2001), we can consider the following kinds of markup:

- markup of educational services for automatic discovery - in this case, markup should be done by providing information relevant to automated classification and selection of educational services (for example, annotation of the service labelled *MIT-intermediary-algebra-course* (belonging to "Learning" category in Figure 8) should make the following relevant information explicit: prerequisites, textbook, term-when-offered, and the like);
- markup for automatic execution of educational services - this means providing information that a pedagogical agent needs in order to construct and execute a service request, to interpret the service's response, and also respond back (input and output

arguments relevant for invoking the function (the program) that implements the service, as well as the language constructs needed to execute the service (such as sequence, iteration, if-then-else));

- markup for automatic composition and interoperation of simple educational services to provide reusable agent procedures - this can be annotated by providing information on prerequisites and consequences of executing each simple service to be integrated in a pedagogical-agent procedure (for example, some explicit logic (rules) to express that "completing course A requires an assessment and lets the student take course B").

Ontological support

Ideally, creation of educational Web contents with ontological annotation should be supported by ontology-driven authoring tools and class hierarchies based on a number of underlying ontologies. Teaching and learning contents of Web-based educational applications can then be presented, edited, modified, and mixed *consistently*. Furthermore, ontologies should be linked to libraries of terms, and interlinked in order to reuse or change terms.

From the author's perspective, the class hierarchies should describe the domain itself, as well as various theories of learning and instructional design process. Of course, nobody expects an authoring tool to be able to support all possible domains and theories, but to support easy access to Web pages (created by other authors) that contain the class hierarchies mentioned, and use them as points of reference.

The reality, however, is still far away from being ideal and there are a lot of further steps and efforts to make in order to move forward. First of all, standard ontologies must be developed to cover different aspects of teaching and learning (e.g., a number of different domains, curriculum sequencing, student modelling, pedagogical issues, grading, and many more). Only a large number of such ontologies will provide the necessary armature for building learning systems on the Web, sharing domain and pedagogical knowledge among the systems, and ensure interoperability and suitable machine interpretation of course material. However, few domain ontologies exist at the moment, and even fewer exist that cover instructional design and learning theories.

One of the reasons why standard ontologies that should cover various areas and aspects of teaching and learning are still missing is the lack of standard vocabulary in the domain of education and instructional design. There are several working groups and efforts towards development of an official standard vocabulary. Examples include the IEEE Learning Technology Standards Committee - http://grouper.ieee.org/groups/ltsc/, Technical Standards for Computer-Based Learning, IEEE Computer Society P1484 - http://www.manta.ieee.org/p1484/, IMS Global Learning Consortium, Inc. - http://www.imsproject.org/, and ISO/IEC JTC1/SC36 Standard - http://jtc1sc36.org/. However, there is still a lot of work to do in that direction. Hence many structural, semantic, and language differences constrain reusability of applications produced by current tools.

Another reason is that current tools for creating Web-based educational applications and those for developing educational ontologies have largely ignored the technological trends, such as the Semantic Web "layer cake". Consider, for example, the most notable work in the AIED community related to the development of educational ontologies, coming from the Mizoguchi Lab at Osaka University, Japan (Mizoguchi, Sinitsa, & Ikeda, 1996; Mizoguchi & Bourdeau, 2000), and from Tom Murray (1998). Ontology-development tools that have resulted from these efforts have implemented a number of important ideas, but did not support XML/RDF encoding of ontologies and consequently were not Semantic Web-ready.

There are three possible ways to go from this current situation, none of them being perfect. First, existing tools for developing Web-based educational applications can be modified to support current Semantic Web languages and semantic markup of resulting documents. This approach would eventually lead to the development of AIED authoring tools for the Semantic Web. The drawback is that such modifications inevitably take time and resources.

Second, we can possibly wait for current authoring tools, such as TopClass, WebCT, Authorware, LearningSpace, CourseInfo, Cyberprof, Mallard, CM Online, and the like, to become more intelligent and more user-friendly, and simultaneously develop suitable plug-ins for ontological support and annotations using, say, DAML+OIL. This is the idea that has been used in the Briefing Associate tool (Tallis et al., 2002). In this case, suitability of each individual authoring tool for such an extension should be judged carefully. Moreover, there is still a competition between Semantic Web languages and there is no guarantee that OWL (or any other language) will win eventually.

The third way is to use existing, though general-purpose Semantic Web tools for ontology development and semantic markup, such as OILed (http://img.cs.man.ac.uk/oil), and Protégé-2000 (http://protege.stanford.edu/). Although not particularly suited for developing *educational* ontologies and knowledge bases, these tools can suffice for the kick-start of development of a number of educational ontologies while some other, possibly better solution appears. Since the vocabulary is not yet standardized officially, only preliminary and incomplete versions of educational ontologies developed by different working groups could result from this approach. The lack of possibility to use terms from official standard vocabulary can be mitigated by using some de facto standards coming from important Internet portals that attract significant numbers of visitors and online transactions. The ontologies themselves would reside on different servers around the Web.

Although it is certainly true that using general-purpose Semantic Web tools is not ideal for developing educational ontologies - such tools lack an instructional design component, to say the least - they are ready, free, and easy to use. For example, the author can use Protégé-2000 to develop important ontologies in a specific domain, Figure 10. Once the ontology tree is developed, the author can use it as the basis for building the domain knowledge of the Web-based learning environment she wants to develop. If she is developing, say, a course unit in one window, another window can be showing the tree of available ontologies. By simply selecting an ontology from that tree as a basis for the unit she is just developing, the author gets a full description of the ontology (its concepts and terms, their relations, the constraints, and possible links to other ontologies) in the third window. Upon saving the author's work, the tool inserts pointers to the ontologies being used into the Web page of the application automatically. Now the course material can be truly distributed over many pages and on different servers, yet all of the pages will be semantically interconnected through the network of ontologies and the courseware developed for the application will be reusable.

Educational services on the Semantic Web

Typical categories of educational services shown in Figure 8 are detailed to an extent in Table 1. The Learning category is rather general and encompasses all services that support the learning process directly. It could certainly be divided into a number of subcategories (authoring, teaching, administration), but the point is that all (sub)categories of services have their distinct educational purpose, properties, and effects. It is exactly these features that must be properly marked-up to make each educational service ready-to-use by pedagogical agents.

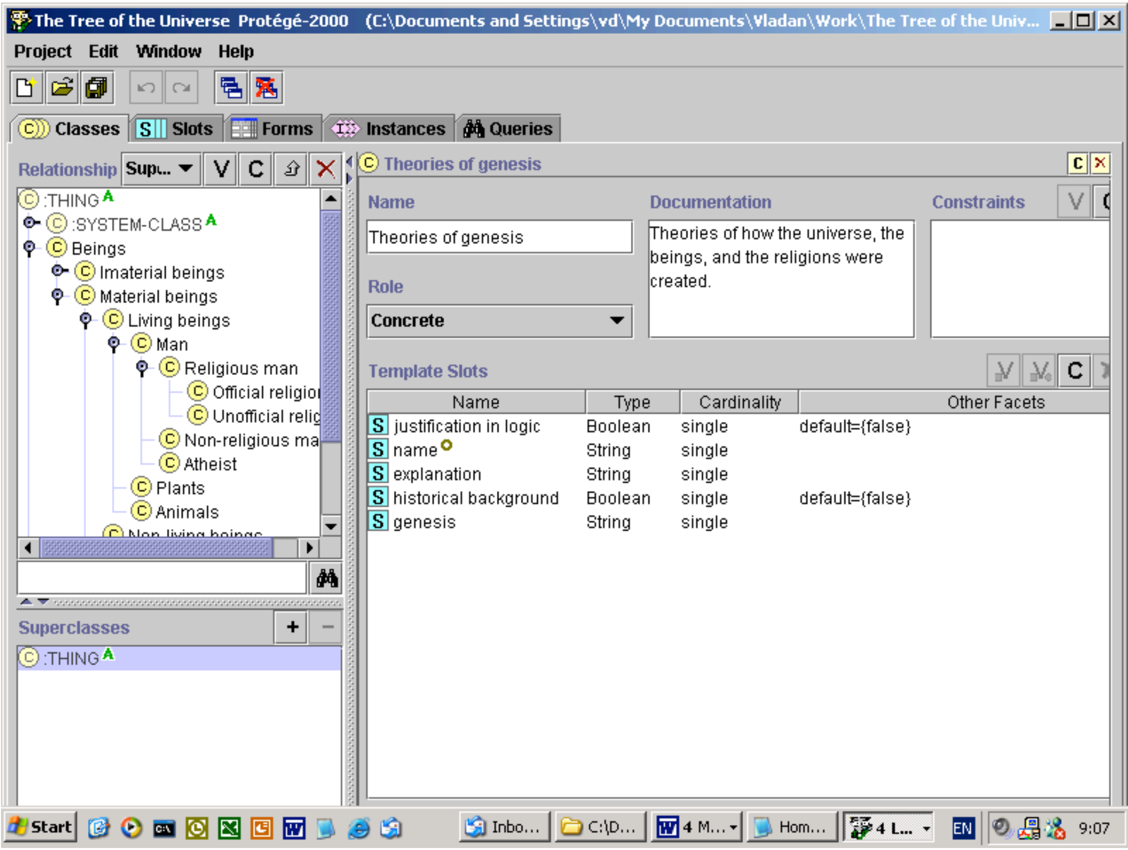


Fig. 10. Working with Protégé-2000

In order to exemplify interoperation between pedagogical agents and educational services on the Semantic Web, consider the following hypothetical scenario. A learner wants to check her competence in computer skills she has learned. She might want to use an assessment service, such as The European Computer Driving Licence® (or ECDL, http://www.ecdl.co.uk/), Figure 11. ECDL is European-wide qualification which enables people to demonstrate their knowledge and skills in the domain of computer technology and use. Suppose, however, that the user doesn't know about the existence of ECDL.

Table 1
Partial classification of educational services

Service category	Learning	Assessment	References	Collaboration
Services	Course offering, integration of educational material, (creating lessons, merging contents from multiple sources, course sequencing), tutoring, presentation	On-line tests, performance tracking, grading	Browsing, search, libraries, repositories, portals	Group formation and matching, class monitoring

Provided that the learner has access to an agent-supported learning environment ready to interact with the Semantic Web, she might want to use her personal agent to arrange the assessment for her. Knowing the learner's profile and goals, the agent will try to discover ECDL and other similar services automatically. The success will depend, of course, also on existing ontological support and on whether ECDL and the other services are suitably marked-up. Assuming that such pre-conditions are satisfied (which is not yet the case in reality), the agent will use ontology-enhanced search engines and pre-provided semantic markup of the services' Web pages and will find the services eventually. In doing so, it may well collaborate and interoperate with other pedagogical agents (see Figure 8).

The agent will then reason about the service(s) discovered and may decide that ECDL is appropriate for its owner. Before showing the ECDL tests to the learner, the agent will use its semantic markup as a declarative API that specifies what input is necessary to execute the service automatically, what information will be returned, and how to actually invoke – and potentially interact with – the service automatically. That may involve automatic service composition and interoperation, in terms of creating a procedure that first registers the user to ECDL (supplying the user's personal data and filling the registration form automatically on behalf of the learner), then collecting the learner's authentification data generated by the registration service (for possible future re-use), then selecting the suitable test level for the learner (see Figure 11), and finally invoking the test service for that level and displaying it to the learner. Alternatively, the learner may have instructed the agent just to find and display relevant information first, without registering automatically. The agent may reason that the procedure to create is "access-the-tests; find-sample-questions; select-the-knowledge-area; select-sample-tests-for-the-knowledge-area". The result may be a sequence of two pages displayed to the learner, Figures 12 and 13. Again, semantic markup of services at their site provides the necessary information for the pedagogical agent(s) to select, compose, and respond to services without much of the learner's intervention. For example, each of the major four links in Figure 11 is a service that should be annotated accordingly for the pedagogical agents to access and interpret them easily.

Although not realistic at the moment, the above example gives a flavour of what kind of services the learner may expect from the Semantic Web. It is hard to say at the moment how long it might take before such a scenario becomes viable, but the good news is that some teams have already started practical developments in that direction.

Fig. 11. ECDL home page

Research, standardization efforts, systems, and practical projects

A word of warning first: everybody is still more-or-less at the beginning. Fortunately, previous work of several groups and individuals has at least paved the road to a good starting point in further developments.

The members of the Mizoguchi Lab, Osaka University, Japan, have developed task ontology for intelligent educational systems (Mizoguchi et al., 1996), an ontology-aware authoring tool (Chen, Hayashi, Kin, Ikeda, & Mizoguchi, 1998), a good theoretical foundation for ontological engineering of educational systems (Mizoguchi & Bourdeau, 2000), as well as several other, practical, working ontologies and ontology-based systems (Mizoguchi & Kitamura, 2001). Further developments in the direction of putting their work into the Semantic Web "layer cake" context would certainly make it even more attractive.

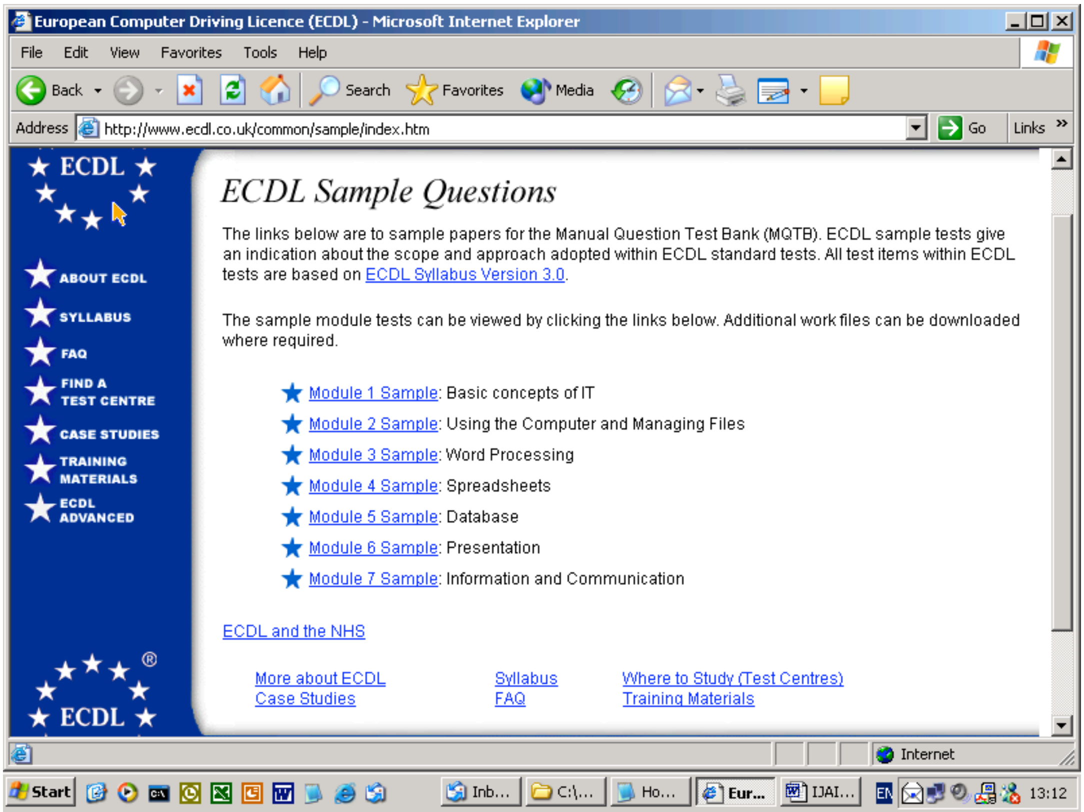


Fig. 12. ECDL sample tests by knowledge areas

The mission of IEEE Learning Technology Standards Committee working groups (LTSC, http://ltsc.ieee.org/) is "to develop Technical Standards, Recommended Practices, and Guides for software components, tools, technologies and design methods that facilitate the development, deployment, maintenance and interoperation of computer implementations of education and training components and systems." Running under a series of projects collectively called P1484, their efforts are related to many important issues of teaching and learning systems, such as architectures, glossary of terms, course sequencing, learner modelling, data interchange protocols, tool/agent communication, and so forth. Of particular interest for education in the context of the Semantic Web are their projects P1484.12, Learning Objects Metadata, and P1484.14, Semantics and Exchange Bindings. The objective of P1484.12 is to "specify the syntax and semantics of Learning Object Metadata, defined as the attributes required to fully/adequately describe a Learning Object." The concept of Learning Object is fairly general, and can include multimedia content, instructional content, learning objectives, instructional software and software tools, as well as persons and organizations. Learning Objects are supposed to be used in technology-supported learning, including computer-based training systems, interactive learning environments, intelligent computer-aided instruction systems, distance learning systems, and collaborative learning environments. The statement of purpose of the project is very detailed, and includes issues like search, evaluation, acquisition, and utilization of Learning Objects, sharing, exchange, composition, and decomposition of Learning Objects across any technology supported learning systems and applications, enabling pedagogical agents to automatically and dynamically compose personalized lessons for an individual learner, enabling the teachers to express educational content in a standardized way, and many more. All of this is actually the essence of teaching and learning on the Semantic Web. P1484.14 supports P1484.12 by proposing and developing techniques such as rule-based XML coding bindings for data models. Finally, it should be noted that such efforts are related to more general standard proposals for ontology development. People involved with the IEEE SUO (Standard Upper Ontology) project 1600.1 (http://suo.ieee.org) are trying to specify an upper ontology that will enable computers to utilize it for applications such as semantic interoperability (not only the interoperability among software and database applications, but also the semantic interoperability among various object-level ontologies themselves), intelligent information search and retrieval, automated inferencing, and natural language processing. They have already proposed the *Information Flow Framework* (IFF) to accomplish the goal of semantic interoperability, as well as the goals related to intelligent services, automated reasoning, and application areas.

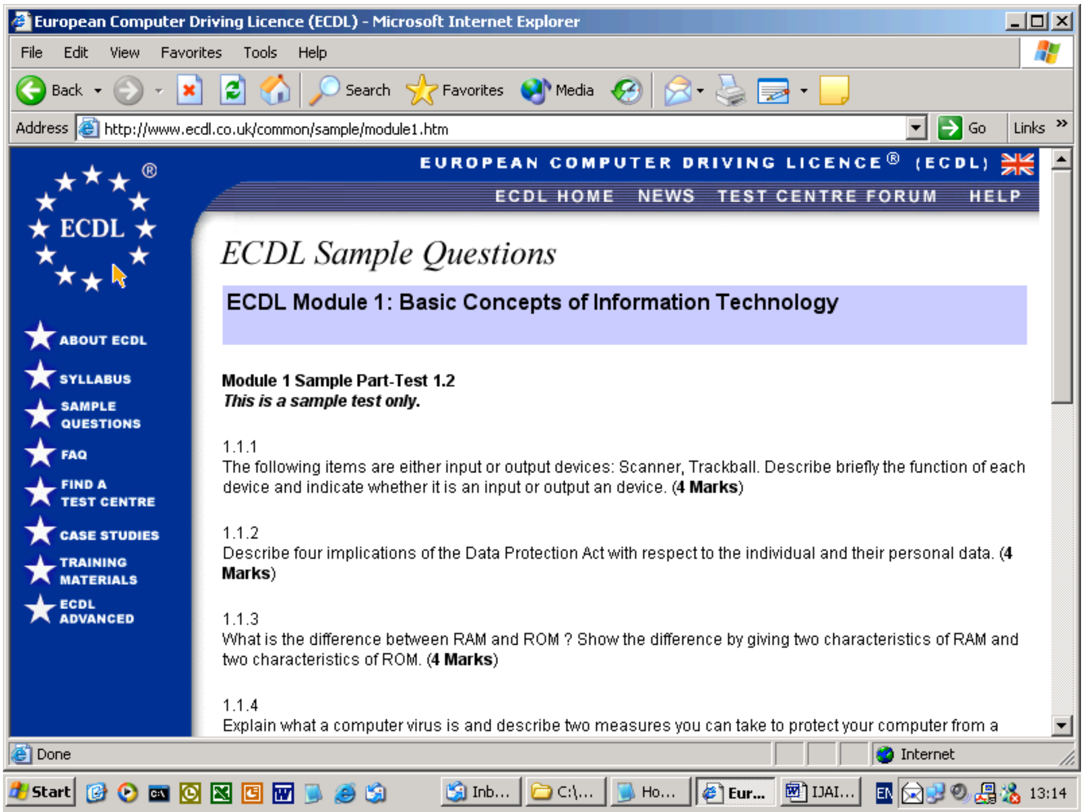


Fig. 13. ECDL sample test in *Basic concepts of IT*

An example of an existing educational site resembling the idea of educational servers as depicted in Figure 8 is GEM (the Gateway to Educational Materials), http://www.geminfo.org. Started as a U.S. Department of Education initiative, GEM is a teacher-oriented educational portal that "expands the educator's capability to access Internet-based lesson plans, curriculum units and other educational materials" by providing "The Gateway" to well-organized, quality collections of various educational resources related to different fields of study. GEM does not use ontologies yet, but makes good use of metadata (specified in RDF), such as *title*, *description*, *grade levels*, *resource type*, and so on.

Practical development of a Semantic Web-based educational application seems to be initiated within the *Universal project* (http://www.ist-universal.org/). The project's objective is to set up an open repository of learning resources on the Web and use it to establish an infrastructure for exchange of activities and collaboration among faculty members of European institutions of higher education. In fact, the repository is supposed to facilitate an open exchange of *learning resources* among participating parties. A learning resource is a form of highly specialized academic content, such as a short video, a simulation, or even a complete course. It is described in terms of its title, objectives, method of instruction, contributors, and curriculum information. A learning resource is generally composed of several *learning objects* (which are associated with physical resources). For example, a learning resource can be a series of tutorials talking about the same topic, each tutorial having a specific format, being associated with specific media, and/or being allocated on a specific server. The project aims at cataloguing and delivery of both live educational sessions and packaged content through the UNIVERSAL Brokerage Platform (UBP), Figure 14. An example of a learning resource in the catalogue is shown in Figure 15.

UBP represents learning objects in the repository starting from the IEEE Learning Object Model (LOM). UBP learning objects are not strictly identical to IEEE LOM learning objects, because UBP introduces some new attributes and modifies some of those proposed by LOM. The implementation of such learning objects and resources is based on RDF and RDF Schemas, many of which are available from the project's site. For example, Figure 16 shows their RDF Schema for learning resources. But in spite of the fact that the Universal project seems to ride on the right track, it also seems to be in the beginning phase.

A noticeable "new wave" of AIED R&D activities related to the Semantic Web started in 2002. Abraham and Yacef (2002) experimented with their XML Tutor in delivering personalized instruction when domain ontology is represented in XML. Cimolino and Kay (2002) presented a system that supports students in creating concept mapping tasks intended to capture the student’s understanding of the ontology of a small domain, as well as to infer his/her misconceptions in the learning process. SITS (Scrutable Intelligent Teaching System) deals with the problem of different understandings (of different authors) of what is most important and how things are related within the domain, i.e. with the existence of different ontologies underlying the sets of teaching documents created by different authors (Kay & Holden, 2002). The approach used to handle this problem is the automatic extraction of the ontology from teaching documents metadata, which are kept separate from the documents. Apted and Kay (2002) went one step further by building a system that automatically constructs an extensive ontology of computer science starting from FOLDOC, the Free On-Line Dictionary of Computing, and using it as a basis for making inferences about student models and other reasoning.

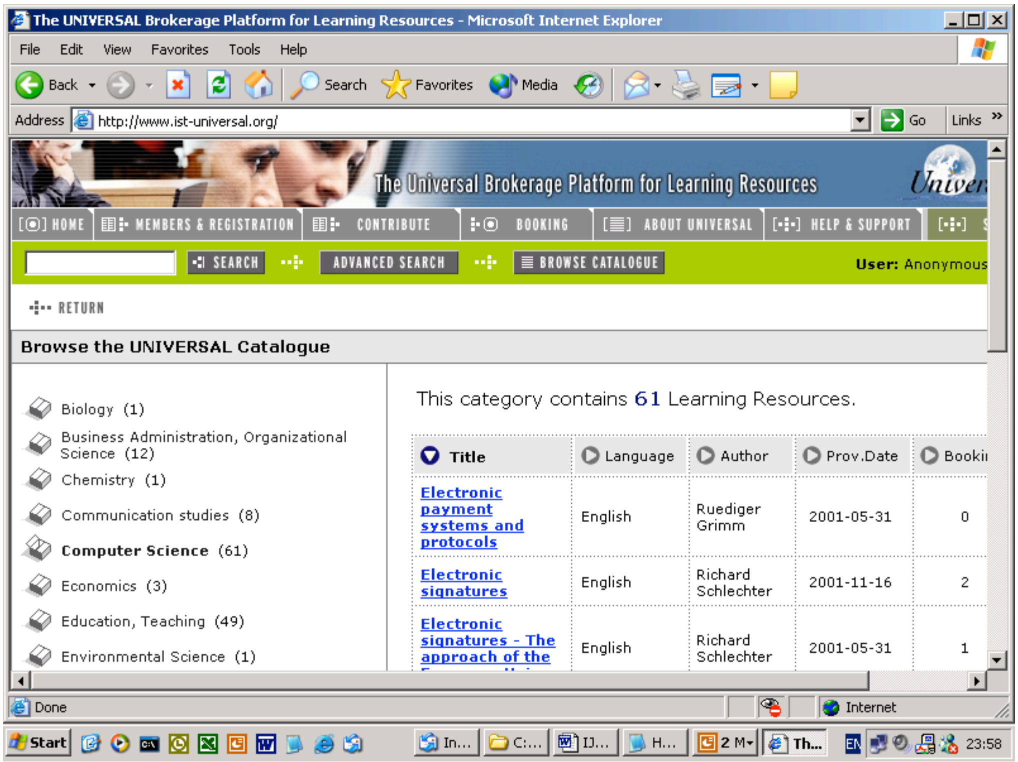


Fig. 14. UNIVERSAL Brokerage Platform (UBP)

Kassist is a workbench for planning problem solving workflow (Seta & Umano, 2002). It takes into account an important difference between the models of problem solving processes and learning processes, and is based on an ontology for enhancing the learners' meta-cognition of their work. Sicilia et al. (2002) introduced the concept of a learning link, as a context-independent, typed entity that can be used to represent (possibly imprecise) semantic relationships between learning resources on the Web. Examples of good engineering design of ontological support for Web courseware authoring include the recently ontology-enhanced AIMS architecture (Aroyo, Dicheva, & Cristea, 2002), and the Ontology Editor (Bourdeau & Mizoguchi, 2002) that enables collaborative ontological engineering involving both a domain expert and an instructional-design expert.

Some of the most recent research in using Semantic Web in educational settings is reported by Gasevic and Devedzic (2004) and Damjanovic et al. (2003). Gasevic has developed the ontology of Petri Nets and used it in a Web-based classroom to support cooperative learning of Petri Nets. Figure 17 shows the idea. Students can use different software tools to develop Petri Nets, but currently different tools support some different features. However, the Petri Nets ontology facilitates automatic exchange of common features Petri Nets models between different tools. In problem-solving tasks, two or more students cooperatively develop the common features using different development tools and exchange the resulting models using the Petri Nets ontology. Then they add specifics supported by individual tools only. A model-exchange Web service is developed for the Web classroom to facilitate this kind of learning.

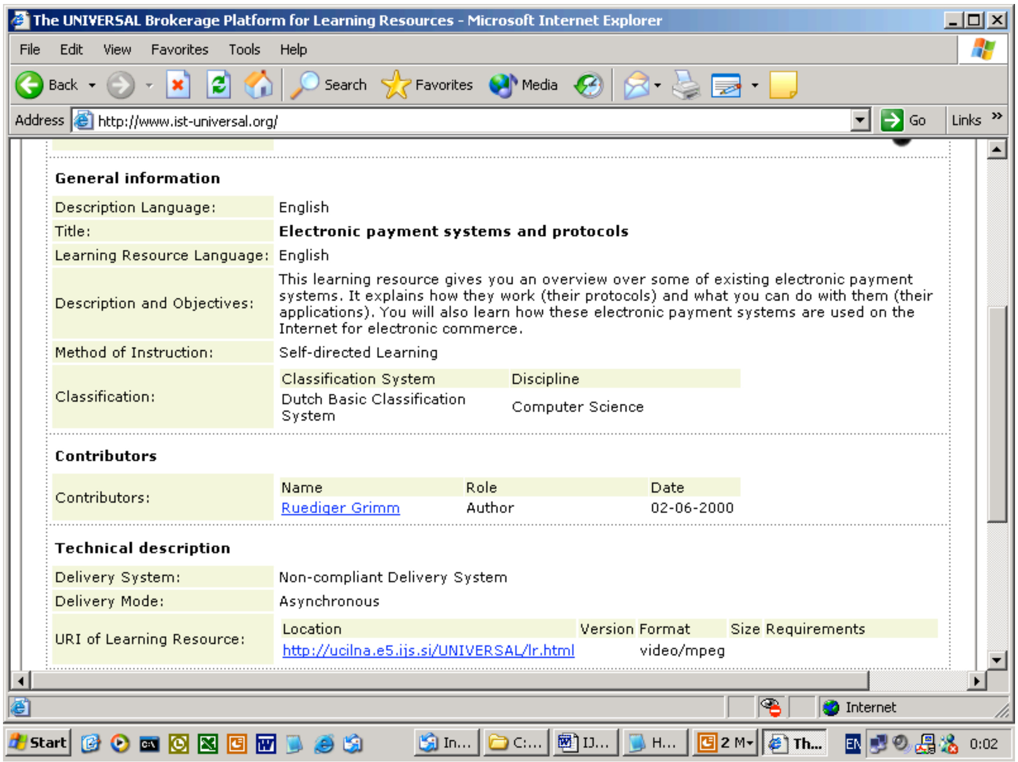


Fig. 15. A learning resource in UBP

Damjanovic et al. (2003) explored a somewhat different idea – can ontology development process *itself* be used to facilitate learning? Developing an ontology is hard work, and usually involves more than one person. They naturally collaborate and learn from each other during the development, and very often their learning motivation is increased. The results of an experiment in that sense, conducted during the development of Damjanovic's ontology of saints and philosophers (part of which is shown in Figure 10), were very encouraging. What is required in such a learning process is a good workbench, or a suite of tools, integrating both learning environments and ontology development tools.

One final remark regarding practical developments: there is little experience so far, and hence little discussion in the literature on *what it really takes* to develop ontologies and *what kind of technology and tools* really provide at least partial semantic interoperability of educational contents on the Web. Some guidelines can be found in (Devedzic, 1999; Devedzic, 2002).

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
  <rdf:Property ID="instructionalDesign">
    <rdfs:label>instructionalDesign</rdfs:label>
    <rdfs:comment>Provides information about the instructional (pedagogical)
        design of the Learning Resource</rdfs:comment>
    <rdfs:range rdf:resource="#InstructionalDesign" />
  </rdf:Property>
  <rdfs:Class rdf:ID="InstructionalDesign">
    <rdfs:label>Instructional Design</rdfs:label>
    <rdfs:comment>Instances of this class represent instructional
        designs</rdfs:comment>
  </rdfs:Class>
  <InstructionalDesign rdf:ID="DirectedLearning" />
  <InstructionalDesign rdf:ID="SelfDirectedLearning" />
  <InstructionalDesign rdf:ID="CollaborativeLearning" />
  <rdf:Property ID="locationOfAdditionalInformation">
    <rdfs:label>locationOfAdditionalInformation</rdfs:label>
    <rdfs:comment>Location (URI) of additional description of the Learning
        Resource.</rdfs:comment>
  </rdf:Property>
  <rdf:Property ID="curriculum">
    <rdfs:label>curriculum</rdfs:label>
    <rdfs:comment>The principal environment (curriculum) within which the learning
        and use of this resource is intended to take place</rdfs:comment>
  </rdf:Property>
  <rdf:Property ID="prerequisite">
    <rdfs:label>prerequisites</rdfs:label>
    <rdfs:comment>Requirements (qualifications) that are needed for this
        LR</rdfs:comment>
  </rdf:Property>
  <rdf:Property ID="learningObject">
    <rdfs:label>learningObject</rdfs:label>
    <rdfs:comment>Link to corresponding LO descriptions</rdfs:comment>
  </rdf:Property>
  <rdf:Property ID="annotation">
    <rdfs:label>annotation</rdfs:label>
    <rdfs:comment>Location of Annotations</rdfs:comment>
  </rdf:Property>
  <rdf:Property ID="offer">
    <rdfs:label>offer</rdfs:label>
    <rdfs:comment>Location of Offer(s) corresponding to LR.</rdfs:comment>
  </rdf:Property>
</rdf:RDF>
```

Fig. 16. UNIVERSAL project: RDF Schema for learning resources

DISCUSSION

True, the AIED field does not seem to have moved significantly forward because of the Semantic Web (yet), nor has it yet demonstrated much synergy with Semantic Web research. The most likely explanation for this fact is that everything in the development of Semantic Web in general, and in its use in education in particular, is only at the beginning. Hence, quite naturally, there is a number of open issues. For example, given a certain concept/topic/theme to learn about from a Web-based interactive learning environment, should there be one large ontology of that concept/topic/theme, or a number of small, inter-related ones? Also, how many ontologies should exist for the same thing? In other words, should everybody be allowed to develop an ontology? If not, who will be granted permission, and who will grant it? Who owns an ontology? How long does it take before people develop the ontology of education as a vertical domain? Since it will probably be a system of many ontologies, not just one, who will be in charge of granting access rights to an educational ontology? Are educational ontologies supposed to live and go through versions somewhere in a large repository (or repositories), or everywhere on the Web?

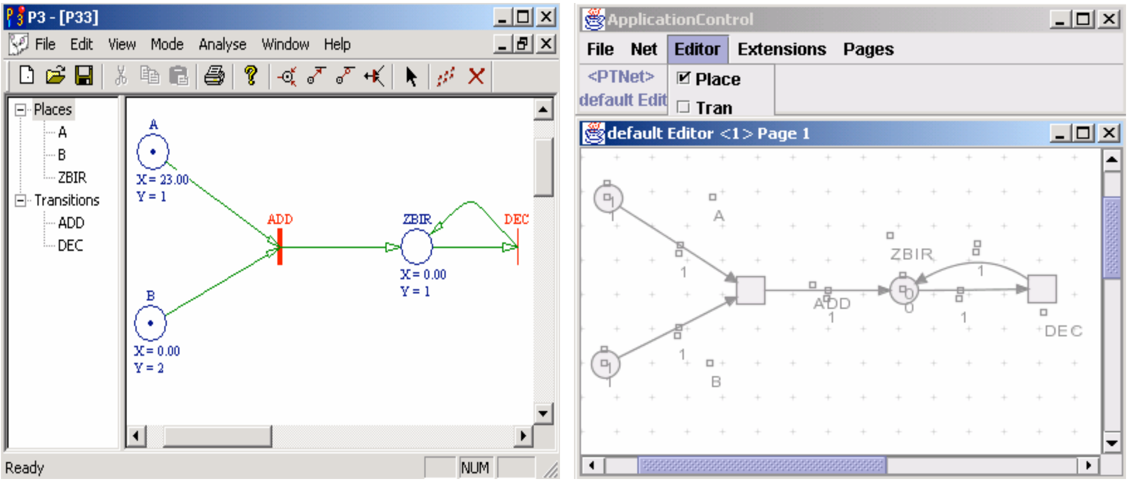


Fig. 17. Learning Petri Nets design cooperatively, using P3 tool (left) and PNK tool (right) to develop the same model

Whatever will be the answers to the above questions, one central problem remains - the actual development of many ontologies. If we are to use the Semantic Web in education, then obviously not only various domain ontologies are needed, but it is also time to start developing educational ontologies as well. Educational ontologies should cover all important concepts and procedures of teaching and learning, as defined in the theories of instruction. While various standardization groups are already making some efforts in that direction, one question we may ask ourselves is: should we wait for such groups to complete their work, and only then start developing Semantic Web-based educational systems, according to the standards? If not, we may be developing in vain, if yes, we don't know how much longer it will take before they finish their work.

One possible way out of this dilemma is to let ontologies *gradually evolve*, while acquiring some experience working with them and while learning more about them. In other words, we can start from some small, largely incomplete educational ontologies, and let them grow incrementally and iteratively over time, as opposed to working on an elaborated, complex conceptual design of ontologies for a long period of time before actually deploying them. This way we want to avoid the "analysis paralysis", i.e. the danger of just thinking about something forever, without putting it to life (Devedzic, 2002). Note, however, that in both cases we must try to match ontologies to the standards still under development.

An important research trend in the Semantic Web community that may support the idea of gradually evolving educational ontologies as well is *ontology learning* (Maedche & Staab, 2001). The idea is to enable ontology import, extraction, pruning, refinement, and evaluation, giving the ontology engineer coordinated tools for ontology modelling. Ontology learning can be from free text, dictionaries, XML documents, and legacy ontologies, as well as from reverse engineering of ontologies from database schemata.

CONCLUSIONS

In developing interactive learning environments, AIED researchers have already adopted a number of general design and development trends. Examples include Web-based systems, open systems, collaborative systems, and adaptive systems, to name but a few. Now that the Semantic Web is apparently just about to come, it is probably the right time to start thinking about adopting it as well. A good thing here is that some AIED researchers have already acquired experience in ontological engineering, one of the key enabling factors for building Semantic Web applications. The danger is failing to recognize what the other factors are, since it may result in just thinking creatively, but having little practical success. Very often, the way out of the maze of many ideas that never come to actual use in practice is to a) start from a well-established technology b) then follow trends and developments in other fields, and c) then apply them to the field of interest. In this case, the field of interest is AI supported teaching and learning on the Web, the other field to look at is ontological engineering, and the technology to start from is the one already developed under the auspices of the WWW Consortium - XML and RDF. If one of the goals of AIED is to build practical systems for the learners and the teachers, then available technology *does* matter.

ACKNOWLEDGEMENTS

This work was supported in part by the Ministry of Science and Technology of the Republic of Serbia, under Grant TR-0004.

REFERENCES

Abraham, D., & Yacef, K. (2002). XMLTutor – an Authoring Tool for Factual Domains. In L. Aroyo, & D. Dicheva (Eds.) Proceedings of ICCE 2002 Workshop on Concepts and Ontologies in Web-based Educational Systems (pp. 7-10). Auckland, New Zealand.

Apted, T., & Kay, J. (2002). Automatic Construction of Learning Ontologies. In L. Aroyo, & D. Dicheva (Eds.) Proceedings of ICCE 2002 Workshop on Concepts and Ontologies in Web-based Educational Systems (pp. 57-64). Auckland, New Zealand.

Aroyo, L., Dicheva, D., & Cristea, A. (2002). Ontological Support for Web Courseware Authoring. In S. Cerri, G. Gouarderes, & F. Paraguacu (Eds.) *Proceedings of The 6th International Conference on Intelligent Tutoring Systems ITS'2002* (pp. 270-280). Biarritz, France and San Sebastian, Spain.

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

Berners-Lee, T., Fischetti, M., & Dertouzos, T. M. (1999). Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by its Inventor. San Francisco: Harper.

Bourdeau, J., & Mizoguchi, R. (2002). Collaborative Ontological Engineering of Instructional Design Knowledge for an ITS Authoring Environment. In S. Cerri, G. Gouarderes, & F. Paraguacu (Eds.) *Proceedings of The 6th International Conference on Intelligent Tutoring Systems ITS'2002* (pp. 399-409). Biarritz, France and San Sebastian, Spain.

Brusilovsky, P. (1999). Adaptive and Intelligent Technologies for Web-based Education. In C. Rollinger & C. Peylo (Eds.) *Künstliche Intelligenz* 4, Special Issue on Intelligent Systems and Teleteaching, 19-25.

Brusilovsky, P., & Miller, P. (2001). Course Delivery Systems for the Virtual University. In F. T. Tschang & T. Della Senta (Eds.) *Access to Knowledge: New Information Technologies and the Emergence of the Virtual University* (pp. 167-206). Amsterdam: Elsevier Science and International Association of Universities.

Chen, W., Hayashi, Y., Kin, L., Ikeda, M., & Mizoguchi, R. (1998). Ontological Issues on an Intelligent Authoring Tool. *Proceedings of The ECAI'98 Workshop on Model-Based Reasoning for Intelligent Education Environments*, Brighton, England.

Cimolino, L., & Kay, J. (2002). Verified Concept Mapping for Eliciting Conceptual Understanding. In L. Aroyo, & D. Dicheva (Eds.) Proceedings of ICCE 2002 Workshop on Concepts and Ontologies in Web-based Educational Systems (pp. 11-16). Auckland, New Zealand.

Damjanovic, V., Gasevic, D., & Devedzic, V. (2003, December 22-25). Ontology Validation. In Proceedings of The Sixth International Conference on Information Technology, CIT 2003 (pp. 183-186.) Bhubaneswar, India.

Decker, S., Melnik, S., van Harmelen, F., Fensel, D., Klein, M., Broekstra, J., Erdmann, M., & Horrocks, I. (2000, September/October). The Semantic Web: The Roles of XML and RDF. *IEEE Internet Computing* 4, 63-74.

Devedzic, V. (1999). Ontologies: Borrowing from Software Patterns. *ACM intelligence Magazine* 10(3), 14-24.

Devedzic, V. (2001). Knowledge Modeling - State of the Art. *Integrated Computer-Aided Engineering* 8, 257-281.

Devedzic, V. (2002, April). Understanding Ontological Engineering. *Communications of the ACM 45*, 136-144.

Fensel, D., & Musen, M.A. (2001). The Semantic Web: A Brain for Humankind. *IEEE Intelligent Systems* 16(2), 24-25.

Fensel, D., van Harmelen, F., Horrocks, I., McGuinness, D.L., & Patel-Schneider, P.F. (2001). OIL: An Ontology Infrastructure for the Semantic Web. *IEEE Intelligent Systems* 16(2), 38-45.

Gasevic, D., & Devedzic , V. (2004). Teaching Petri nets using P3. Accepted for publication in *Educational Technology and Society*.

Gómez-Pérez, A., & Corcho, O. (2002). Ontology Languages for the Semantic Web. *IEEE Intelligent Systems* 17(1), 54-60.

Heflin, J., & Hendler, J. (2001). A Portrait of The Semantic Web in Action. *IEEE Intelligent Systems* 16(2), 54-59.

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

Horrocks, I., Fensel, D., Broekstra, J., Decker, S., Erdmann, M., Goble, C., et al. (2002). The Ontology Inference Layer OIL, Tech. Report, Vrije Universiteit, Amsterdam. Retrieved March 19, 2002, from http://www.ontoknowledge.org/oil/TR/oil.long.html

Horrocks, I. & van Harmelen, F. (2002). Reference Description of the DAML+OIL Ontology Markup Language. Retrieved March 19, 2002, from http://www.daml.org/2000/12/reference.html

Johnson, W.L., Rickel, J., & Lester, J.C. (2000). Animated Pedagogical Agents: Face-to-Face Interaction in Interactive Learning Environments. *International Journal of Artificial Intelligence in Education* 11, 47-78.

Kay, J., & Holden, S. (2002). Automatic Extraction of Ontologies from Teaching Document Metadata. In L. Aroyo, & D. Dicheva (Eds.) Proceedings of ICCE 2002 Workshop on Concepts and Ontologies in Web-based Educational Systems (pp. 25-28). Auckland, New Zealand.

Klein, M. (2001). Tutorial: The Semantic Web - XML, RDF, and Relatives. *IEEE Intelligent Systems* 16(2), 26-28.

Lassila, O. (1998). Web Metadata: A Matter of Semantics. *IEEE Internet Computing* 2(4), 30-37.

Maedche, A., & Staab, S. (2001). Ontology Learning for the Semantic Web. *IEEE Intelligent Systems* 16(2), 72-79.

McIlraith, S.A., Son, T.C., & Zeng, H. (2001). Semantic Web Services. *IEEE Intelligent Systems* 16(2), 46-53.

Mizoguchi, R., Sinitsa, K., & Ikeda, M. (1996). Task Ontology Design for Intelligent Educational/Training Systems,. In Proceedings of the Workshop "Architectures and Methods for Designing Cost-Effective and Reusable ITSs" (pp. 1-20). Montreal, Canada.

Mizoguchi, R., & Bourdeau, J. (2000). Using Ontological Engineering to Overcome Common AI-ED Problems. *International Journal of Artificial Intelligence in Education* 11, 1-12.

Mizoguchi, R., & Kitamura, Y. (2001). Knowledge Systematization Through Ontology Engineering - A key technology for successful intelligent systems. Invited paper at PAIS 2001, Seoul, Korea, November 2001. Retrieved December 21, 2001, from http://www.pais2001.org/english.htm

Murray, T. (1998). Authoring Knowledge-Based Tutors: Tools for Content, Instructional Strategy, Student Model, and Interface Design. *The Journal of the Learning Sciences* 7(1), 5-64.

Noy, N. F., Sintek, M., Decker, S., Crubézy, M., Fergerson, R.W., & Musen, M.A. (2001). Creating Semantic Web Contents with Protégé-2000. *IEEE Intelligent Systems* 16(2), 60-71.

Preece, A., & Decker, S. (2002). Intelligent Web Services. *IEEE Intelligent Systems* 17(1), 15-17.

Scott Cost, R., Finin, T., Joshi, A., Peng, Y., Nicholas, C., Soboroff, I., et al. (2002). ITtalks: A Case Study in the Semantic Web and DAML+OIL. *IEEE Intelligent Systems* 17(1), 40-47.

Seta, K., & Umano, M. (2002). A Support System for Planning Problem Solving Workflow. In L. Aroyo, & D. Dicheva (Eds.) Proceedings of ICCE 2002 Workshop on Concepts and Ontologies in Web-based Educational Systems (pp. 29-35). Auckland, New Zealand.

Sicilia, M.A., García, E., Díaz, P., & Aedo, I. (2002). LEARNING LINKS: Reusable Assets with Support for Vagueness and Ontology-based Typing. In L. Aroyo, & D. Dicheva (Eds.) Proceedings of ICCE 2002 Workshop on Concepts and Ontologies in Web-based Educational Systems (pp. 37-42). Auckland, New Zealand.

Swartout, W., & Tate, A. (1999). Ontologies, Guest Editors' Introduction. *IEEE Intelligent Systems* 14(1), 18-19.

Tallis, M., Goldman, N.M., & Balzer, R.M. (2002). The Briefing Associate: Easing Authors into the Semantic Web. *IEEE Intelligent Systems* 17(1), 26-32.

Wuwongse, V., Anutariya, C., Akama, K., & Nantajeewarawat, E. (2002). XML Declarative Description: A Language for the Semantic Web. *IEEE Intelligent Systems* 17(1), 54-65.