Scientific Annals
of the „Alexandru Ioan Cuza” University of Iaşi
Economic Sciences
60 (1), 2013, 187-198
DOI 10.2478/aicue-2013-0005

A SURVEY OF WEB ONTOLOGY LANGUAGES AND SEMANTIC WEB SERVICES

Alexandru Napoleon SIRETEANU
Faculty of Economics and Business Administration
Alexandru Ioan Cuza University
Iasi, Romania
salexis_ro@yahoo.com

Abstract

In the beginning World Wide Web was syntactic and the content itself was only readable by humans. The modern web combines existing web technologies with knowledge representation formalisms. In this sense, the Semantic Web proposes the mark-up of content on the web using formal ontology that structure essential data for the purpose of comprehensive machine understanding. On the syntactical level, standardization is an important topic. Many standards which can be used to integrate different information sources have evolved. Beside the classical database interfaces like ODBC, web-oriented standard languages like HTML, XML, RDF and OWL increase in importance. As the World Wide Web offers the greatest potential for sharing information, we will base our paper on these evolving standards.

Keywords: Ontologies, Cycl, KIF, RDF, OWL, DAML and OIL, Semantic Web Services
JEL classification: L63, L86

1. INTRODUCTION

The World Wide Web is composed of HTML documents that can be characterized as a syntactic or visual web because documents are only displayed by web browsers. In the visual web, machines cannot understand the meaning of the information present in HTML pages, since they are mainly made up of ASCII codes and images. The visual web prevents computers from automating information processing, integration, and interoperability.

The developed technologies that made the concept of the semantic web possible are presented in Table 1. As we can see, the web was initially a vast set of static web pages (HTML files) linked together, that are still used by many organizations. Due to the dynamic nature of businesses, organizations are using dynamic publishing methods which offer great advantages over web sites constructed from static HTML pages. Server-side applications and database access techniques are used to dynamically create web pages directly in response to requests from user browsers and offer the opportunity to deliver web content that is highly customized to the needs of individual users. The technologies available to dynamically create web pages based on database information were insufficient for the requirements of organizations looking for application integration solutions. The Extensible

Markup Language (XML) was one of most successful solutions developed to provide business-to-business integration. XML-based solutions for applications and systems integration were not sufficient, because the data exchanged lacked an explicit description of its meaning. Semantic integration and interoperability is concerned with the use of explicit semantic descriptions to facilitate integration. The vision behind of the symbiotic web is interaction between humans and machines in synergy that it will be possible to build more powerful interfaces such as mind controlled interfaces using web 4.0. In simple words, machines would be clever on reading the contents of the web, and react in the form of executing and deciding what to execute first to load the websites fast with superior quality and performance and build more commanding interfaces (Hemnath, 2010).

Technologies/ Characteristics	Static	Dynamic	Syntax	Semantic	Symbiotic
Coding	HTML	PHP, MySQL	XML	RDF/OWL/OIL	Atoms
Creation	Manually or using web page authoring tools	By server-side applications	By application based on schema	By applications based on models	By sensors
Users	Humans	Humans	Humans and applications	Humans and applications	Objects and applications
Paradigm	Browse	Create/Update/ Delete queries	Integrate	Interoperate	Object Hyperlinking
Applications	Browsers	Browsers	Process Integration, Workflows	Intelligent agents, Semantic engines	Intelligent web, webOS
	1995	2000		2006	2012

Source: [the author]

Figure no. 1 Evolution of Web technologies

Currently the web is undergoing an evolution and different approaches are developed for adding semantics to web pages and resources in general. Semantic Web merges together a network of information that allows more efficiency, greater knowledge sharing and ease of use. Ontologies are the key to this interoperability because they determine the language that software agents need to communicate with each other and humans need to communicate with the agents.

Due to the widespread importance of integration and interoperability for intra- and inter-business processes, the research community has already developed several semantic standards such as the resource description framework (RDF), RDF schema (RDFS) and the Web Ontology Language (OWL). RDF, RDFS and OWL standards enable the web to be a global infrastructure for sharing both documents and data, which make searching and reusing information easier and more reliable. RDF is a standard for creating descriptions of information, especially information available on the World Wide Web. OWL provides a language for defining structured Web-based ontologies which allows a richer integration and interoperability of data among communities and domains.

Ontologies were developed in Artificial Intelligence to facilitate knowledge sharing and become common in fields such as intelligent information integration, information retrieval, knowledge management and education. The reason why ontologies have become popular is due to a shared and common understanding of some domain that can be communicated between people and application systems. Ontologies are formal theories about a certain domain of communication and require formal logical languages for describing ontologies.

2. ONTOLOGY LANGUAGES

The word "ontology" has its origin in philosophy and refers to all elements within a a system and the relationships between them. For example, the observation that the world is made of specific objects that can be grouped into abstract classes based on shared properties is a typical ontological statement. The challenge of designing ontology languages can be described as a compromise between the expressive power needed to approximate human conceptualization, and the minimal complexity needed to achieve a practical result in the target application area.

There are three essentials ontology languages which are popular and prototypical for a specific language paradigm:

- *CycL* and *KIF* as illustratives of improving first-order predicate logic languages;
- *Ontolingua* and *Frame Logic* as representatives of frame-based methods. Both techniques integrate frame-based modeling primitives in a first-order logical framework, but they apply different strategies for this;
- *Description Logics* that describe knowledge in terms of concepts and role restrictions used to automatically extract classification taxonomies.

CycL is a formal language whose syntax is derived from first-order predicate calculus and extends first-order logic through the use of second order concepts. The vocabulary of CycL consists of terms: semantic constants, non-atomic terms, variables, numbers, strings, etc. Terms are combined in CycL expressions, ultimately forming closed CycL sentences. A set of CycL sentences forms a knowledge base (Cycorp, Inc., 2012). In brief, CycL uses predicate logic extended by typing (e.g. functions and predicates are typed), reification (e.g. predicates and formulas are treated as terms and can be used as expressions within other formulas) and microtheories that define a context for the truth of formulas.

The *Knowledge Interchange Format* (KIF) is a language designed for use in the exchange of knowledge between distinct computer systems, created by different programmers at different times, in different languages. Being a language for knowledge interchange, KIF can also be used as a language for expressing and exchanging ontologies. The language has declarative semantics, is logically complete and provides an instrument for the representation of knowledge about knowledge. This allows the user to make knowledge representation decisions explicit and to introduce new knowledge representation constructs without changing the language (Knowledge Interchange Format, 2012).

KIF and CycL are both oriented on predicate logics and provide an important extension of first-order logic. They allow the reification of formulas as terms used in other formulas and meta-level statements. In consequence, CycL is a modeling language for ontologies whereas KIF was designed as an exchange format for ontologies, and both languages are close in spirit to RDF.

Frame-based and object-oriented techniques have a different point of view relative to modeling primitive of predicate logic. Their predominant modeling primitive are classes with specific properties called attributes.

Ontolingua was designed to support the design and specification of ontologies with a clear logical semantics based on KIF. Ontolingua extends KIF using additional syntax to include the intuitive collection of axioms into definitional forms with ontological significance and a Frame Ontology to define object-oriented and frame-language terms. The set of KIF expressions that Ontolingua allows is defined in an ontology called the Frame Ontology. The Frame Ontology specifies the representation primitives that are often supported by special-purpose syntax and code in object-centered representation systems (for example classes, instances). Ontolingua ontology is made up of definitions of classes, relations, functions, objects, and axioms that describe these terms (Ontolingua, 2012).

Frame logic is a language for specifying object-oriented databases, frame systems, and logical programs. The main goal of this language is to integrate conceptual modeling constructs (classes, attributes, domain restrictions, inheritance, axioms) into a consistent logical framework. In essence it provides classes, attributes with domain and range definitions, hierarchies with set inclusion of subclasses and multiple attribute inheritance, and logical axioms that can be used to characterize the relationships between elements of an ontology and its instances (Frame Logic, 2012).

Ontolingua and Frame logic integrate frames into a logical framework. The main difference between Ontolingua and Frame logic is the fashion in which they realize frame-based modeling primitives in a logical language. Ontolingua characterizes the frame-based modeling primitives via axioms in the language. Frame logic defines their semantics externally via an explicit definition of their semantics.

Description Logics, also known as terminological logics, forms an important powerful class of logic-based knowledge representation languages. A distinctive property of Description logics are classes that can be defined deliberate in terms of descriptions and specify the properties that objects must satisfy in order to be the property of the concept. These descriptions are expressed using a language that allows the construction of composite descriptions, including restrictions on the binary relationships connecting objects (Description Logics, 2012).

3. RDF - A DATA-MODEL FOR META-INFORMATION

XML language was designed to provide an interchange format for inadequate structured data by defining a data model in a schema and using annotations from the schema in order to relate information items to the schema specification. The best thing that XML offers for ontological modeling is the Document Type Definition (DTD) which defines the nesting of tags and introduces attributes for them. Defining tags, their nesting, and attributes for tags may be seen as defining an ontology.

RDF is a framework for describing web resources (identified by Uniform Resource Identifier or URIs) such as homepage, title, author, content and copyright information of a web page. RDF is a data model for objects ("resources") and relations between them. RDF provides a simple semantics for this data model, and this can be represented in XML syntax. RDF is designed to be read and understood by computers and describes resources with properties and property values.

The RDF standard has proposed a simple data model for representing meta-data about web pages and their content using XML syntax. In this model, every type of information about a resource, which may be a web page or an XML element, is indicated in terms of a triple (Resource Description Framework, 2012): (resource, property, value)

The property is a relation that connects a resource to a specific value of that resource. A value can be a simple data type or a resource and can be replaced by a variable representing a resource that is in addition described by linking triples making assertions about the properties of the resource that is represented by the variable:

(resource, property, X)
(X, property_1, value_1)
...
(X, property_n, value_n)

RDF can be used directly to describe an ontology. Objects, classes, and properties can be described. Predefined properties can be used to model instance of and subclass of relationships as well as domain restrictions and range restrictions of attributes. Those properties are defined globally and are not encapsulated as attributes in class definitions. A frame or object-oriented ontology can only be expressed in RDF by reifying the property names with class name suffixes (like in XML).

RDF is an application of XML used for representing metadata. For example, the RDF statements:

date(http://www.feaa.uaic.ro/Example/Alex/) = April 2007
subject(http://www.feaa.uaic.ro/Example/Alex/) = Ontology Languages
creator(http://www.feaa.uaic.ro/Example/Alex/) = http://www.hostname1.net/~Alex/
name(http://www. feaa.uaic.ro/~Alex/) = Alex
email(www.feaa.uaic.ro/~Alex/) = alex@feaa.uaic.ro

can be represented in linear XML syntax (see Figure 2). The difference between them is that RDF provides a standard form for representing metadata in XML. Directly using XML to represent metadata would establish various ways of representation. The difference is more relevant when we try to represent an ontology in RDF or XML.

```
<? xml version=”1.0”?>
<RDF
xmlns=”http://www.w3c.org/1999/02/22-rdf-
syntax-ns#”
xmlns:DC=”http://www.purl.org/DC#/”
xmlns:y=”http://www.description.org/schema”>
<Description about=”
http://www.feaa.uaic.ro/Example/Alex/”>
<DC:date rdf:resource=” April 2007”/>
<DC:subject rdf:resource=”Ontology
Languages”/>
<DC:creator rdf:resource=”
http://www.feaa.uaic.ro/~Alex/”/>
</Description>
<Description about=”
http://www.feaa.uaic.ro/~Alex/”>
<DC:name rdf:resource=”Alex”/>
<DC:email rdf:resource=” alex@feaa.uaic.ro”/>
</Description>
</RDF>
```

Source: [the author]

Figure no. 2 XML representation of RDF statements

RDF provides a fixed set of modeling primitives for defining an ontology (classes, resources, properties, is-a and element-of relationships) and a standard way to encode them in XML. Using XML directly for the purpose of representing ontologies would require maximizing the standardization effort.

Case Study. Implementing an RDF Schema for the economic journal Scientific Annals of the "Alexandru Ioan Cuza" University of Iasi (Economic Sciences Section) for indexing the journal in RePEc (Research Papers in Economics)

All RDF metadata should be available on a web server in order that the RePEc service can retrieve them regularly to integrate them into their databases. The folder that is holding these data is our RePEc archive. There are two ways to do this: using a web server or using an anonymously FTP server. RePEc recommends the second solution because it is more effective, the server is not overloaded and there aren't any security problems. In some cases, it is impossible to properly prepare the server, because it would violate certain local policies related to IT (RePEc, 2012).

RePEc archive can be located anywhere on the server, as long as it is accessible via web or anonymous ftp; recommended directory name could be RePEc / AIC / journal where aic is the archive code for the "Alexandru Ioan Cuza University" of Iasi. Structure rules about the archive are quite strict (see Figure 3).

- RePEc/
    - aic/
        - aicarch.rdf
        - aicseri.rdf
        - journl/
            - RDF files for articles
        - wpaper/
            - RDF files for working papers

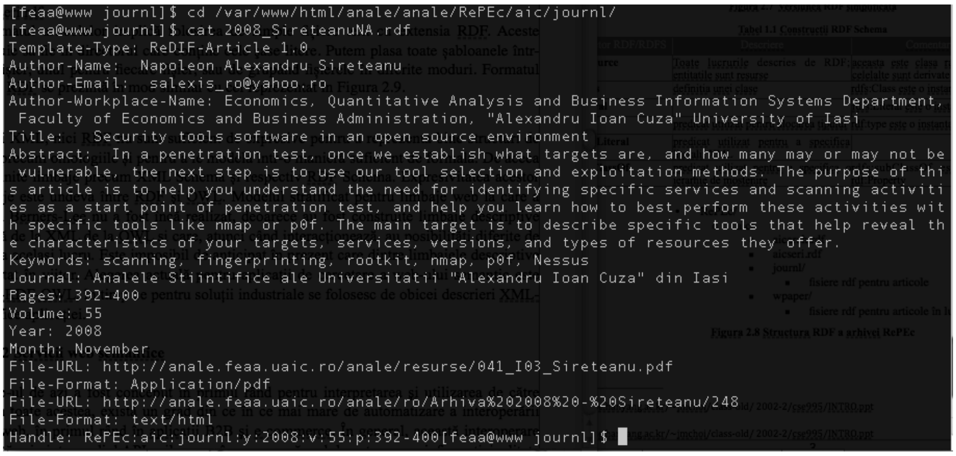
Source: [the author]

Figure no. 3 RDF structure for RePEc archive

The two files from the aic directory describe the archive; aicarch.rdf contains technical details, including archive URL, while aicseri.rdf describes the content of the archive, i.e. series of articles that are in progress or completed items for which we will provide data. For each series or journal a subdirectory with a name that matches with the information from aicseri.rdf file should be present.

Articles format involves using simple text files with the RDF extension. These files must be in the directory that contains the six letters. We can put all templates in a single file, one for each file, or group the files in various ways. The RDF format for an article is similar to that shown in Figure 4.

Neither XML nor RDF are enough expressive to represent knowledge structures such as ontologies and to modeling them in a formal manner. For this reason languages like XML Schema and RDF Schema were created. The expressiveness of these two languages is somewhere between RDF and OWL.

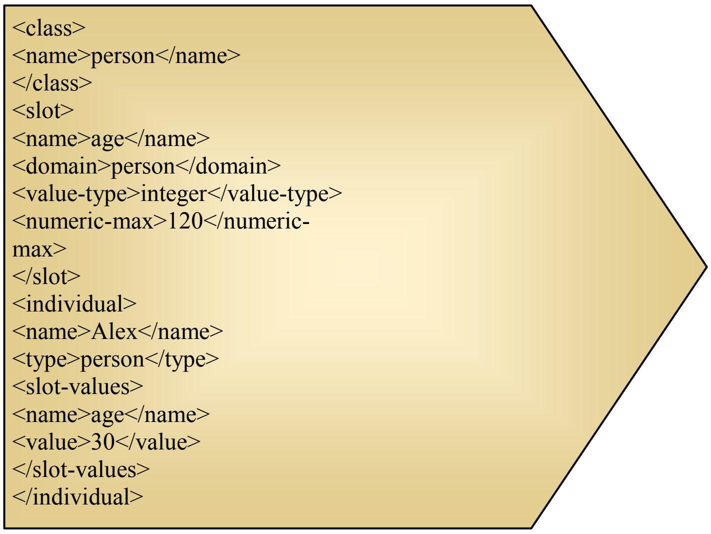


Source: [the author]

Figure no. 4 RDF article from RePEc archive for the journal Scientific Annals of University "Alexandru Ioan Cuza" University (Economic Sciences Section)

4. A COMPARATIVE APPROACH OF THE SEMANTIC WEB LANGUAGES

Except the two standards examined above, a number of other techniques for encoding ontologies on the World Wide Web have been proposed. A comparison of some specialized languages, including RDF schema and DAML+OIL will be presented in the next paragraphs.



Source: [the author]

Figure no. 5 An example in XOL

In comparison to RDF schema which has been presented above, there are languages that support knowledge representation on the web and are compatible with web standards XML or RDF:

- *XOL* (XML-based Ontology Language) has been proposed as a language for exchanging formal knowledge models in the domain of bio-informatics. The development of XOL has been managed by the representational needs of the domain and by existing frame-based knowledge representation languages. The design of XOL uses a generic approach to define ontologies, meaning that the single set of XML tags defined for XOL (defined by a single XML DTD) can describe any ontology. As we can see in Figure 5, all of the XML elements of this specification, such as class, individual, and name are generic and apply to all ontologies.

The fundamental advantage of the XOL technique is simplicity. Only one XML DTD needs to be defined to describe any ontology. Using the non-generic approach, every ontology must define a second, ontology-specific, DTD for describing the data elements of the ontology. Rules would have to be defined for describe exactly how that second DTD is derived from the schema portion of the ontology (XOL Ontology Exchange Language, 2012).

- *SHOE* (Simple HTML Ontology Extension) was created as an extension of HTML for the purpose of defining machine readable semantic knowledge. The purpose of SHOE is to enable intelligent web agents to retrieve and gather knowledge more precisely than it is possible in the presence of plain HTML documents;

- *OML* (Ontology Markup Language) is an ontology language that has initially been developed as an XML serialization of SHOE. The semantics for the higher levels is based on the idea of conceptual graphs;

- *OIL* (Ontology Inference Layer) is an effort to develop an ontology language for the Web that has a well defined semantics and sophisticated logic support for ontology development and use. The language provide a formal semantics for RDF schema, standard-OIL which is equivalent to a description logic, an Instance OIL that adds the possibility of defining instances and a Heavy OIL that permits future extensions (Description of OIL, 2012).

OIL extends RDF schema and has been the main impact in the development of DAML+OIL. The main distinction between OIL and DAML+OIL is an extended meaningful of DAML+OIL for complex definitions of individuals and data types.

Because of the current dominance and importance of the WWW, a syntax of an ontology exchange language must be formulated using existing web standards for information representation. OIL is similar relative to XOL and can be seen as an extension of XOL. RDF provides two important contributions: a standardized syntax for writing ontologies, and a standard set of modeling primitives like instance of and subclass of relationships. For that reason, OIL offers two syntactical variants: one based on XML schema and one based on RDF schema.

- *DAML+OIL* is the successor of OIL, defined in collaboration with research groups from the DARPA, following the original versions of OIL and DAML-ONT (DARPA Agent Markup Language, 2012).

DAML+OIL is similar to OIL in many aspects, but is more tightly integrated with RDF, which provides the only specification of the language and its only serialization. While the dependence on RDF has some advantages in terms of the re-use of existing RDF infrastructure and the portability of DAML+OIL ontologies, using RDF to completely define the structure of DAML+OIL is quite difficult as, unlike XML, RDF is not designed for the precise specification of syntactic structure.

From a practical point of view, DAML+OIL implementations can choose to support some or all of the XML schema data types. For supported data types, they can either implement their own type checker or depend on some external component.

- *OWL* (Web Ontology Language) is the most powerful of the ontology languages currently defined for the Semantic Web. Unlike DAML+OIL, OWL is sponsored by World Wide Web Consortium (W3C) (OWL Web Ontology Language, 2012). OWL has facilities for expressing meaning and semantics and the ability to represent machine interpretable content on the web. OWL is designed for use by applications that need to process the content of information instead of just presenting information to humans.

OWL can be viewed as a collection of RDF triples, but those triples that use the OWL vocabulary have a specific OWL-defined meaning. If a given RDF graph instantiates the OWL specification, then OWL provides a semantic interpretation for the components of that graph or sub-graph (Sireteanu, 2006, pp. 65-80).

The comparison of the languages mentioned above depends of the set of elements contained in the language and their ability to encode semantic information about a domain in respect to the following aspects (Daconta et. al, 2003, pp. 181-263):

- Concepts and Taxonomies: Ontologies group objects that have certain properties in common (e.g. cities or countries). A description of the shared properties is called a concept definition. Concepts can be arranged into a subclass-superclass relation for discriminating objects into sub-groups (e.g. capitals or East-European countries);

- Relations: Objects of the same type appear in similar situations where they have an evident relation to each other (cities are located in countries, countries have a capital). These typical relations can commonly be specified for establish structures between groups of objects;

- Instances: In some cases single objects (e.g. the continent Europe) play a notable role in a domain of interest or the membership to a concept is defined by the relation to a specific object (European countries are those contained in Europe). For this reason ontology languages allow to specify single objects, called instances;

- Axioms: Sometimes a domain move behind apparent rules that cannot be expressed with the elements presented above (e.g. the fact that the number of residents of Europe equals the sum of the number of residents of European countries).

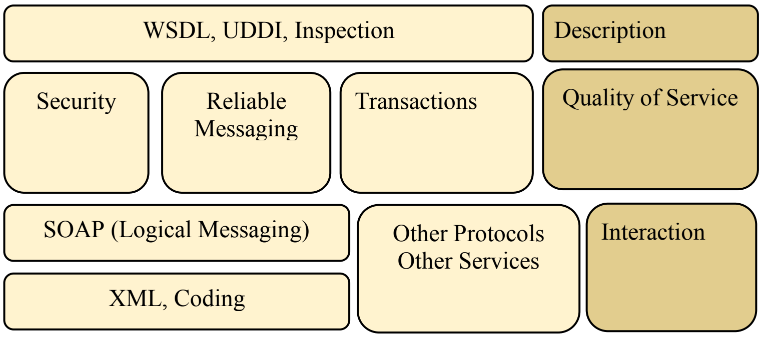
The comparison revealed significant differences in the sense of eloquences of the different languages. It is important to analyze the characteristic needs of a particular application before choosing one of the languages. The result of this analysis should guide the selection of one of the languages examined above. In my opinion DAML+OIL, the predecessor of OWL, is the most significant language for encoding ontologies on the web.

5. WEB SERVICES AND SEMANTIC WEB SERVICES

Web services are technologies that allow programmatic access to resources on the Internet. Web services provide a method to create distributed systems which are loosely couple, interaction between the client and service being not dependent on one having any knowledge of the other. This type of interaction between components is defined formally by the service-oriented architecture (SOA).

Extensible Markup Language (XML) is a platform independent data representation which allows the flexibility of Web Services. Simple object access protocol (SOAP) is the XML-based protocol that governs the communication between a service and the client, and it provides a platform and programming language for Web services to exchange messages. Web Service Description Language (WSDL) is an XML-based language that describes all the information needed to present and use a Web Service. UDDI is a standard for storing WSDL files as a registry so that they can be discovered by clients.

Web Services technology has gained a proper level of maturity and is being used to easily publish business functions to Intranet/Internet for remote execution. Business functions can exist in popular applications such as ERP (enterprise resource planning), CRM (customer relationship management), and SCM (supply chain management) systems. A part of the standards associated with Web services are indispensable to developing SOA based solutions and are represented in Figure 6:



Source: [the author]

Figure no. 6 Web Services and standards

The service oriented architecture (SOA) is seen like an evolution of the distributed systems technology of the 1990s, such as DCOM and CORBA. This type of architecture requires the existence of main components and concepts such as services, service descriptions, service security parameters and constraints, advertising and discovery, and service contracts in order to implement distributed systems. Web Services can be seen as a specialized SOA implementation that represents the core aspects of a service-oriented approach to architecture. With Web Services, developers don't need to know how a remote program works, only the input that it requires, the output it provides and how to use it for execution.

Web Services are modular, self-describing, self-contained applications that are accessible over the Internet. Currently, Web Services are described using the Web Services Description Language, which provide operational information. Web services address the needs of application integration by providing a standards-based framework for exchanging information dynamically between applications. The industry standards (WSDL, UDDI, and SOAP) are designed to represent information about the interfaces of services, how they are deployed, and how to invoke them, but are limited to express the capabilities and requirements of services. Because of deficit of semantic representation capabilities, the automatic integration of applications written to Web Services standards is unsatisfied, and for this reason Semantic Web community has introduced Semantic Web services. Semantic Web Service representation mechanisms are OWL-S, WSMO and WSDL-S.

OWL-S is projected to enable users and software agents to automatically discover, invoke, compose, and monitor Web resources offering services, under specified restrictions.

Web Service Modeling Ontology (WSMO), a European Union funded project, proposes four main elements:

- ontologies, which provide the terminology used by other WSMO elements,
- Web Service descriptions, which describe the functional and behavioral aspects of a Web service,
- Goals that represent user desires, and
- Mediators, which aim at automatically handling interoperability problems between different WSMO elements.

Considering increasing compatibility of WSDL, semantic support for XML schema and industry adoption issues, WSDL-S proposes an incremental approach to add semantic annotations to WSDL documents.

6. CONCLUSIONS

In present, computers are changing from single isolated devices to mediators into a world wide network of information exchange and business transactions. Data, information, and knowledge exchange is becoming the key issue in contemporary computer technology. Ontologies provide a common understanding of a domain that can be communicated between people and different application systems and will play a major role in supporting information exchange processes in various areas. The use of one ontology for all application will never be possible. Never an ontology will be convenient for all subjects and domains or for a large and varied community such as the Web community.

In the future Semantic Web will drive a new era of real world applications. With its ability to support every business domain, computer science and information systems experts have to reconsider their role. They must be able to transform business requirements to systems and solutions that go beyond traditional analysis and design. It is obviously necessary to include the Semantic Web in computer science and information systems curricula of universities. Educating people from computer science departments and business schools about Semantic Web implies making them realize that semantics, logic, reasoning, and trust are just our mankind characteristics that we must bring to our computer world.

Acknowledgment

This work was supported by the project "Post-Doctoral Studies in Economics: training program for elite researchers - SPODE" co-funded from the European Social Fund through the Development of Human Resources Operational Programme 2007-2013, contract no. POSDRU/89/1.5/S/61755.

References

Cycorp, Inc., 2012. [online]. Available at: http://www.cyc.com/ [Accessed 20.02.2012]

Daconta, M., Obrst, L.J. and Smith, K.B., 2003. The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management, Wiley, pp. 181-263.

DARPA Agent Markup Language, 2012. [online]. Available at: http://www.daml.org/language/, [Accessed 9.03.2012]

Description Logics, 2012. [online]. Available at: http://dl.kr.org/ [Accessed 25.02.2012]
Description of OIL, 2012. [online]. Available at: http://www.ontoknowledge.org/oil/, [Accessed 7.03.2012]
Hemnath, 2010. Web 4.0 - A New Web Technology, [online]. Available at: http://website-quality.blogspot.com/2010/01/web-40-new-web-technology.html [Accessed 15.02.2012]
Knowledge Interchange Format, 2012. [online]. Available at: http://www-ksl.stanford.edu/knowledge-sharing/kif/ [Accessed 21.02.2012]
Ontolingua, 2012. [online]. Available at: http://www-ksl-svc.stanford.edu:5915/ [Accessed 22.02.2012]
Frame Logic, 2012. [online]. Available at: http://flora.sourceforge.net/aboutFlogic.php [Accessed 23.02.2012]
Resource Description Framework, 2012. [online]. Available at: http://www.w3.org/RDF/ [Accessed 26.02.2012]
Sireteanu, N., A., 2006. The impact of globalization on economic information systems, PhD. Thesis, pp. 65-80.
OWL Web Ontology Language, 2012. [online]. Available at: http://www.w3.org/TR/owl-features/ [Accessed 9.03.2012]
XOL Ontology Exchange Language, 2012. [online]. Available at: http://www.ai.sri.com/pkarp/xol/ [Accessed 5.03.2012]