



**DESIGN AND IMPLEMENTATION OF AN
EXAMPLE SEMANTIC WEB BASED SYSTEM**

**(ANLAMSAL WEB TABANLI ÖRNEK BİR SİSTEM
TASARIMI VE GERÇEKLEŞTİRİMİ)**

Berna PAKALIN GENÇOĞLU*, Alp KUT*

ABSTRACT/ÖZET

The Semantic Web is envisioned as the future of the current Web. The Semantic Web proposes the mark-up of Web content so that machines can automatically process, interpret and integrate information available on the Web. This study explains the vision, architecture, theory, and fundamental components of the Semantic Web and presents design and implementation of an example prototype Semantic Web based system for an academic department of a university.

Anlamsal Web mevcut Web'in geleceği olarak betimlenmektedir. Anlamsal Web, Web içeriğinin işaretlenmesini ve böylece makinaların Web üzerindeki bilgiyi otomatik işleyebilmesini, yorumlayabilmesini ve birleştirilebilmesini önermektedir. Bu çalışma, Anlamsal Web'in vizyonunu, mimarisini, teorisini, temel bileşenlerini açıklar ve bir üniversitenin akademik bir bölümü için Anlamsal Web tabanlı örnek bir prototip sistem tasarımını ve gerçekleştirimini sunar.

KEYWORDS/ANAHTAR KELİMELEER

Semantic web, Ontology
Anlamsal web, Ontoloji

1. INTRODUCTION

The World Wide Web (WWW) has gained considerable attention and success due to its distributed nature and its ability which allows users to easily access and share information. Currently on the Web, there is a vast repository of information comprising of billions of documents. At the moment, however, the extensive information found on the Web is designed primarily for human consumption. The content is marked up in such a way that data is displayed in a meaningful way to humans. As such the Web content is not meant for machine consumption. HTML (Hyper Text Markup Language) is the predominant language in which web pages are written. It relies on a set of predefined tags which control appearance of a Web page. It only carries information on how the content should be displayed. It provides neither the structure nor the semantics of information. Due to these limitations in the current Web technology, research efforts have been put towards to enhance the current Web in such a way that it will allow the information to be processed more efficiently. Tim Berners-Lee, who invented the WWW in 1989 and is now the director of the World Wide Web Consortium (W3C), introduced the concept of Semantic Web as the future of the current Web. The Semantic Web is not a separate Web but an extension of the current one, in which information is given a well-defined meaning, better enabling computers and people to work in cooperation (Berners-Lee et al., 2001).

The Semantic Web technology promises to dramatically improve the current Web and its use. For the Semantic Web, the word semantic states that the meaning of data on the Web can be understood-not just by people, but also by machines. The Semantic Web is all about making the Web meaningful, understandable, and machine-processable. The Semantic Web is a vision for the future of the Web in which information is organized with semantic mark-up to allow machines to automatically access, process and integrate information available on the Web.

2. SEMANTIC WEB TECHNOLOGY

The development of the Semantic Web is being promoted by the World Wide Web Consortium (W3C). W3C is an international consortium that founded in 1994 for developing guidelines, technologies, specifications, and standards for the Web. Figure 1 shows the layered architecture of the Semantic Web as it is presented by the W3C (W3C, 2003).

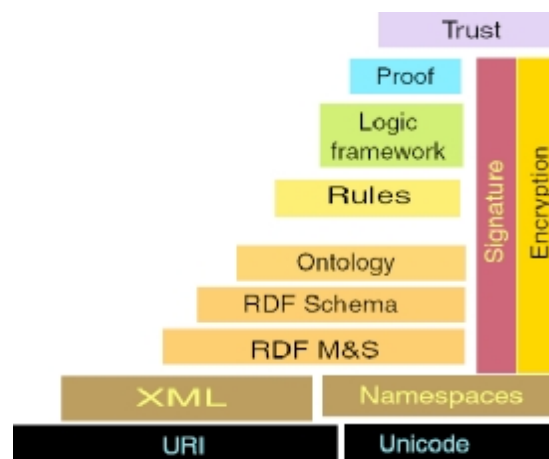


Figure 1. The Semantic Web layered architecture

2.1. Uniform Resource Identifier (URI) and Unicode

The base of architecture is the concept of URI and a universal character set (Unicode). URI provides a naming scheme and is used to identify or name a web resource on the Web. The named resources can be any abstract or physical thing that has an identity. A URI consists of a string of characters and conforms to a particular syntax. URI became standard with the invention of the Web, and it serves as the foundation not only for the current Web but also for the Semantic Web.

Unicode is a universal character encoding standart that is designed to support the worldwide interchange, processing, and display of data in any language. It is a 16 bit character set standard that covers all the characters and symbols of the modern written languages. It is universal and unique and provides an unambiguous encoding of the content of plain text, covering all languages in the world.

2.2. Extensible Mark-up Language (XML) and Namespaces

XML is a simple general purpose mark-up language that is used to create semantically rich Web documents. XML is a subset of SGML, the Standard Generalized Mark-up Language [ISO8879] and has been designed for ease of implementation and for interoperability with both SGML and HTML (Bray et al., 2006). The XML specification was approved as a W3C Recommendation in February 1998. Like an HTML document, an XML document consists of a collection of tagged elements. However unlike HTML which relies on a set of pre-defined tags for formatting and displaying information on a Web page, XML allows users to write structured Web documents with user defined vocabulary. Any number of XML tags can be defined and nested to any level of complexity. XML provides a way of encoding both data and meta-data.

Namespaces are used to allow unambiguous use of several vocabularies. An XML Namespace is a W3C standard for providing unique names for the elements and attributes in an XML document. XML namespace allows a combination of several vocabularies within a single document without ambiguity. Since each vocabulary is given a namespace, duplicate element and attribute names can be distinguished and used without a name conflict.

2.3. Resource Description Framework (RDF)

RDF provides a standard for describing resources. A resource can be any thing that is identified with an URI. Making statements about resources provide descriptions of the resources and a basis for information representation. RDF became a W3C Recommendation in February 1999. RDF is intended for situations in which information needs to be processed by applications, rather than only being displayed to people, and it provides a common framework for expressing information so it can be exchanged between applications without loss of meaning (Manola and Miller, 2004). RDF is based on the idea of expressing simple statements about resources, where each statement is triple comprised of a subject, a predicate, and an object. The subject is the resource being described. The predicate is the relation between the subject and the object. The object is the part that identifies the value of that property. RDF provides an XML-based syntax (called RDF/XML) to represent RDF statements in a machine-processable way.

2.4. RDF Schema

Above the RDF layer, RDF Schema is a framework that provides a means to describe vocabularies of RDF resources. RDF's vocabulary description language, RDF Schema, is a semantic extension of RDF and it provides mechanisms for describing groups of related resources and the relationships between these resources (Brickley and Guha, 2004). Although RDF provides a base for describing resources on the Web, it does not provide means for defining classes and properties. RDF Schema builds on RDF foundation to provide richer representation formalism. RDF Schema allows the definition of classes, subclasses, sub-properties, domain and range restrictions of properties. In other words, RDF Schema introduces basic ontological modelling primitives. RDF Schema became a W3C Recommendation in February 2004.

2.5. Ontology

Ontology layer expands RDF Schema and allows the representation of more complex semantics. RDF Schema is not very expressive since it only has limited modelling primitives. RDF Schema provides a means to define vocabulary, structure and constraints for expressing metadata about web resources, however formal semantics for the primitives defined in RDF Schema are not provided, and the expressivity of these primitives is not enough for full-fledged ontological modelling and reasoning (Broekstra et al., 2000). The limitations of the RDF Schema specification language has led to efforts in designing more powerful ontology languages on top of RDF Schema.

2.6. Rules, Logic Framework, Proof and Trust

Higher levels of the Semantic Web, which are the layers of Rules, Logic Framework, Proof and Trust, are still subject to ongoing research and have not been well defined yet. On top of ontologies, a logic framework and rules are needed to be specified for a fully functional Semantic Web. Logic framework enhances the ontology language further and provides a way to describe and exchange logic assertions over the Web. A rule language allows inference rules to be implemented which allow machines to draw conclusions and make deductions. The proof layer executes the rules, and traces and explains the steps of logical reasoning. Lastly the trust layer provides evidence of the trustworthiness of data, services and agents. The trust layer clarifies whether and/or how much the information present on the Semantic Web can be trusted. Digital signature and encryption provides a basis for the trust layer and envisioned to be used on all levels of the Semantic Web. Digital signature and encryption across the layers ensures the authentication, integrity, nonrepudiation, and confidentiality of data.

3. ONTOLOGIES

The building of a Semantic Web application essentially relies on ontologies since they give a formal model of the domain by explicitly describing concepts and their relations in some logical formalism. Ontologies allow defining sets of terms as well as relations that exist between these terms in a structured and unambiguous way. The formalism makes an ontology machine interpretable so that ontologies can be put into computation and inference by machines.

Many disciplines now develop ontologies that experts can use and share information within their fields. Some of the important reasons for developing ontologies can be summarized as follows:

- to define a vocabulary within a domain.
- to provide common understanding of information in a domain.
- to share common understanding of the structure of information among people or software agents.
- to enable machines to use knowledge in some application.
- to enable reuse of domain knowledge.

There is much literature on what ontologies are, how they can be implemented, what tool suits are, and where they can be used inside applications. All these topics and activities are covered under the term “Ontological Engineering”. Ontology engineering refers to the set of activities that concern the ontology development process, the ontology life cycle, the methods and methodologies for building ontologies, and the tool suits and languages that support them (Gomez-Perez et al., 2004).

3.1. Definition of Ontology

The term ontology originates from Philosophy. In that context, ontology is a systematic explanation of existence. It tries to explain what exists by introducing a system of basic categories. Later, in the 1990s, the term ontology became a topic of interest in computer science, especially in Artificial Intelligence community for knowledge sharing and reuse. Many definitions on what ontology is have been proposed in literature.

Neches et al. defines ontology as the basic terms and relations comprising the vocabulary of a topic area as well as the rules for combining terms and relations to define extensions to the vocabulary (Neches et al., 1991). Gruber defines ontology as an explicit specification of a conceptualisation (Gruber, 1993). This definition became the most quoted in the literature. Based on Gruber’s definition many definitions of ontology were proposed. Gruber’s definition was explained by Studer, Benjamins, Fensel as follows: “An ontology is a formal, explicit specification of a shared conceptualisation (Studer et al., 1998). A ‘conceptualisation’ refers to an abstract model of some phenomenon in the world by having identified the relevant concepts of that phenomenon. ‘Explicit’ means that the type of concepts used, and the constraints on their use are explicitly defined. ‘Formal’ refers to the fact that the ontology should be machine readable, which excludes natural language. ‘Shared’ reflects the notion that an ontology captures consensual knowledge, that is, it is not private to some individual, but is accepted by a group”. Guarino and Giaretta collected and classified seven definitions and proposed to interpret ontology as a logical theory which gives an explicit, partial account of a conceptualization (Guarino and Giaretta, 1995).

3.2. Ontology Development

Ontologies are used to represent the background knowledge that is needed by the application. Currently there is no standard ontology building methodology. Different methodologies are developed based on project characteristics. Noy and McGuinness states that there is no one “correct” way or methodology for developing ontologies and offers a very simple possible process based on iterative design that can be used in developing ontologies (Noy and McGuinness, 2001). The main sequence of the steps is listed as follows:

- Determine the domain and scope of the ontology

- Consider re-using existing ontologies
- Enumerate important terms in the ontology
- Define the classes and class hierarchy
- Define the properties of classes – slots
- Define the facets of the slots
- Create instances

3.3. Ontology Languages

Ontology languages are quite essential and needed for describing a model into concrete, computer serializable files. Ontology languages provide some forms of logics and formal modelling means considering the strictness of expression and ease of computation. An ontology language must fulfil three important requirements (Fensel et al., 2001):

- It must be highly intuitive to the human user. Given the current success of the frame-based and object-oriented modelling paradigm, an ontology should have a frame-like look and feel.
- It must have well-defined formal semantics with established reasoning properties to ensure completeness, correctness, and efficiency.
- It must have a proper link with existing Web languages like XML and RDF to ensure interoperability.

Many Ontology markup languages have been created in context of the Semantic Web. Figure 2 shows the stack of these ontology markup languages: Simple HTML Ontology Extension (SHOE), Ontology Exchange Language (XOL), Ontology Markup Language (OML), RDF Schema, Ontology Inference Layer (OIL), DARPA Agent Markup Language+OIL (DAML+OIL), and Web Ontology Language (OWL) (Gomez-Perez and Corcho, 2002).

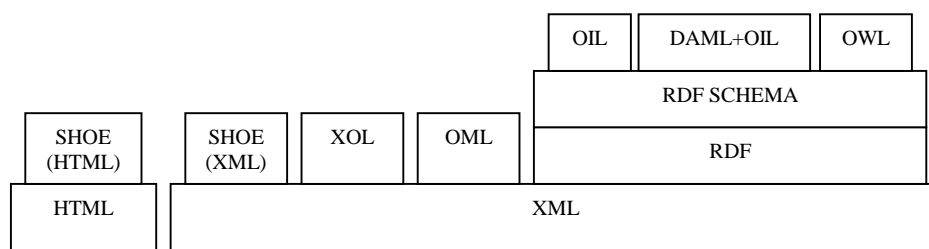


Figure 2. Language stack for the Semantic Web

OIL, DAML+OIL, and OWL have been developed as extensions to RDF Schema.

OIL is a Web-based representation and inference layer for ontologies. It combines the widely used modelling primitives from frame-based languages with the formal semantics and reasoning services provided by Description Logics (Fensel et al., 2000).

DAML+OIL is an ontology language specifically designed for the use on the Web; it exploits existing Web standards (XML and RDF), adding the familiar ontological primitives of the object oriented and frame based systems, and the formal rigor of a very expressive description logic (Horrocks, 2002). DAML+OIL is the result of merging Ontology Inference Layer (OIL) and DARPA Agent Markup Language (DAML). Later DAML+OIL is submitted to W3C as a prototype of Web Ontology Language (OWL) which is now the W3C standard ontology language.

OWL is a revision of the DAML+OIL web ontology language incorporating lessons learned from the design and application of DAML+OIL (McGuinness and van Harmelen, 2004). OWL aims to provide sufficient expressive support to describe machine-processable content needed for Semantic Web applications. OWL is the proposed standard for building ontologies on the Web. OWL became a W3C Recommendation in February 2004, taking its place alongside other Web standards such as URI, XML etc.

3.4. Ontology Development Tools

Tools are important and needed for the purposes of ontology development and manipulation. There are numerous tools developed to support and ease the task of ontology development. Gomez-Perez collects and introduces a wide range of ontology tools by categories as follows (Gomez-Perez, 2002) :

Ontology development tools: This group includes tools that can be used to build a new ontology from scratch. In addition to common editing and browsing functionality, these tools usually provide ontology documentation, ontology export/import to/from different formats and ontology languages, ontology graphical edition, ontology library management, etc.

Ontology merge and integration tools: These tools are used to solve the problem of merging and aligning different ontologies in the same domain.

Ontology evaluation tools: They are used to evaluate the content of ontologies and their related technologies. Ontology content evaluation tries to reduce problems that can be encountered on integrating and using ontologies and ontology-based technology in other information systems.

Ontology-based annotation tools: These tools allow users to insert instances of concepts and of relations in ontologies and maintain (semi)automatically ontology-based markups in Web pages.

Ontology storage and querying tools: These tools allow using and querying of ontologies easily and performing inferences with them.

Ontology learning tools: They are used to (semi)automatically derive ontologies from natural language texts, and semi-structured sources and databases, by means of machine learning and natural language analysis techniques.

4. A SEMANTIC WEB BASED SYSTEM

In this study, an example Semantic Web based system for an academic department of a university has been designed and implemented. The system extracts information from ontology and makes it accessible to the end-user. Users can access the system through a conventional Web browser interface.

4.1. Architecture of the System

The overall architecture of the system consists of three main layers: User Interface Layer, Internal Layer, and Knowledge Layer. This architecture is illustrated in Figure 3.

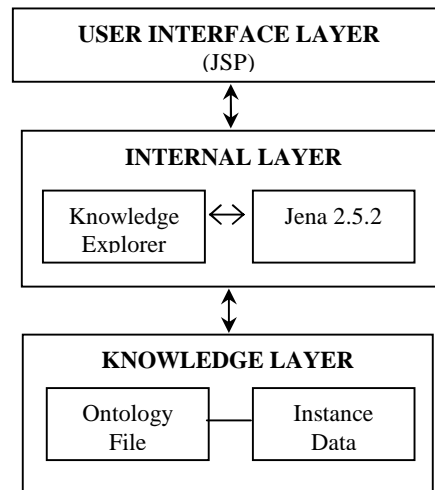


Figure 3. Overall architecture of the system

User Interface Layer interacts with the user and handles all incoming requests submitted by the users. The users can get the results and explore the ontology information using a conventional Web browser.

Internal Layer handles the interaction with knowledge base and runs the business logic. Knowledge Explorer located at that layer gets the ontology and instance data from Knowledge Layer and transforms them to triples using Jena 2.5.2 API as the parser. Content is dynamically generated and presented to the user in HTML interface layout.

Knowledge Layer includes either a local or a remote file system that hosts the knowledge base file which contains the ontology and instances data in it.

4.2. Ontology of the System

Development of the ontology constitutes a core task of implementing a Semantic Web based system. With ontologies, the information is denoted in a semantically machine-processable format which enables machines to deduce additional information, a task difficult in a traditional database system. W3C recommendation, OWL is selected as the ontology development language due to its expressiveness, standard, widespread recognition, and tool support. Classes, properties and instances are formally represented with OWL language in the ontology file. In the development of the ontology, a popular ontology editor, Protégé 3.2 with OWL plug-in, is used. Protégé is used basically to construct the ontology in OWL and also for the purpose of knowledge-acquisition, i.e. instantiating the classes defined in the ontology. Protégé is a free, open source, graphical ontology editor that is built in Java. It is being developed by Stanford University and is supported by a strong community of developers. Protégé has an easy-to-use graphical user interface and an extensible plug-in architecture. The Protégé-OWL plug-in extends the Protégé platform into an ontology editor for the OWL.

There are twelve classes defined in the ontology of the system. The hierarchy of classes are listed below, where a subclass is indented to show the subclassOf.

```

Owl:Thing
  Courses
  Publication
  ResearchTopics
  
```


Staff
 AcademicStaff
 Asistant
 Instructor
 Professor
 NonAcademicStaff
 Students
 Graduate
 UnderGraduate

The Ontology of the system includes thirty-two properties. Table 1 summarizes the properties defined in the ontology of the system. Figure 4 depicts the graphical representation of the ontology implemented. It displays all the classes, properties of the classes and the relationships between the classes.

Table 1. List of properties defined in the ontology

RELATION	TYPE	DOMAIN	RANGE
Assists	Object property	Assistant	Courses
hasAssistant	Object property	Courses	Assistant
hasClassNumber	Datatype property	UnderGraduate	String
hasCourseDescription	Datatype property	Courses	String
hasCourseHomePage	Datatype property	Courses	String
hasCourseID	Datatype property	Courses	String
hasCourseName	Datatype property	Courses	String
hasCredits	Datatype property	Courses	String
hasDegree	Datatype property	Graduate	String
hasEmail	Datatype property	Staff	String
hasFirstName	Datatype property	Staff	String
hasHomepage	Datatype property	AcademicStaff	String
hasInterest	Object property	Academic Staff	ResearchTopics
hasJobTitle	Datatype property	NonAcademicStaff	String
hasLastName	Datatype property	Staff	String
hasPhoneNumber	Datatype property	Staff	String
hasPublicationName	Datatype property	Publication	String
hasPublicationType	Datatype property	Publication	String
hasPublicationYear	Datatype property	Publication	String
hasStudentID	Datatype property	Students	String
hasStudentName	Datatype property	Students	String
hasStudentSurname	Datatype property	Students	String
hasTitle	Datatype property	Professor	String
hasTopic	Object property	Publication	ResearchTopics
hasTopicName	Datatype property	ResearchTopics	String
interestedBy	Object property	ResearchTopics	AcademicStaff
isPublishedBy	Object property	Publication	AcademicStaff
isTakenBy	Object property	Courses	Students
isTaughtBy	Object property	Courses	Professor, Instructor
Publishes	Object property	AcademicStaff	Publication
Takes	Object property	Students	Courses
Teaches	Object property	Professor, Instructor	Courses

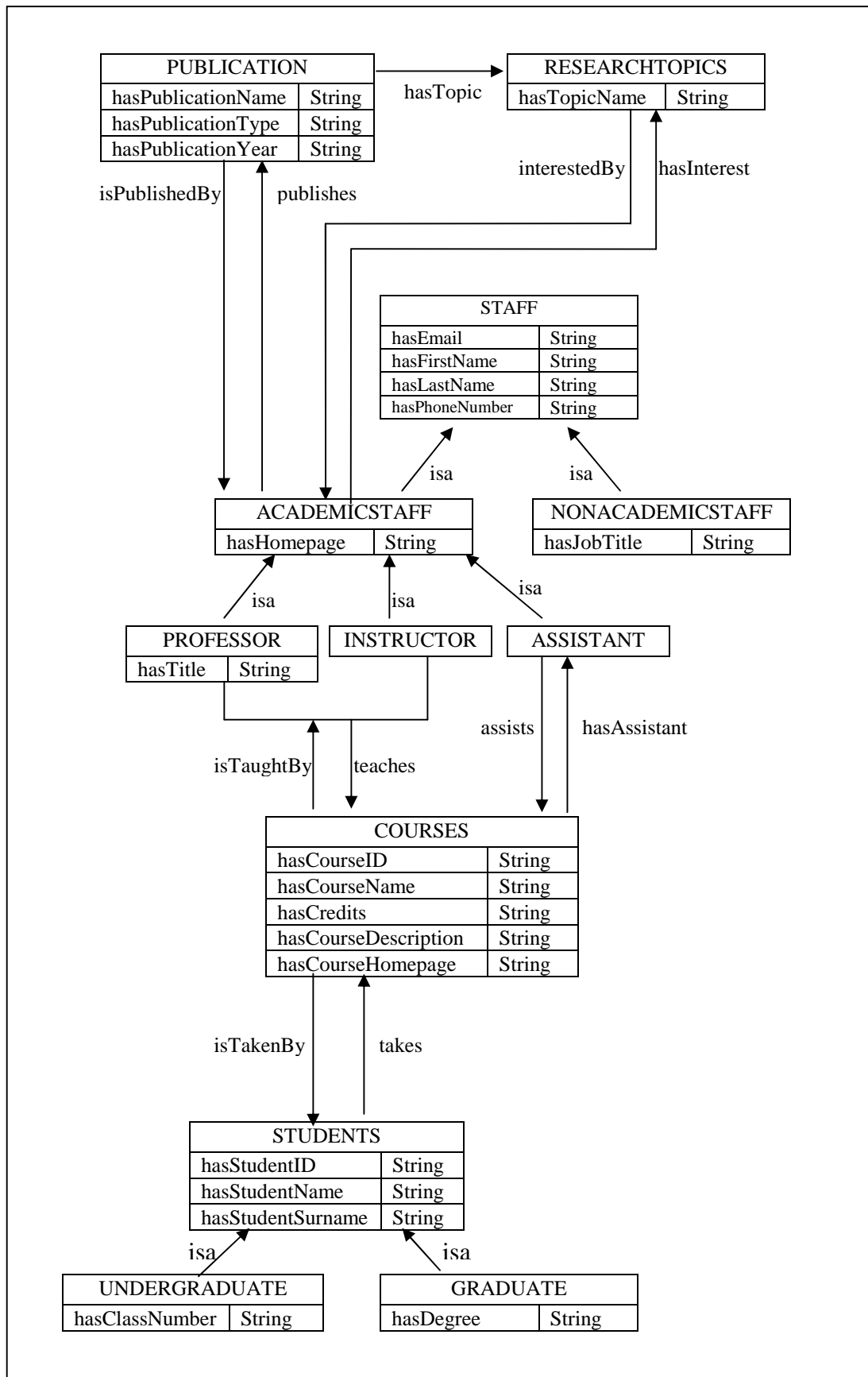


Figure 4. Graphical representation of the ontology

4.3. Programming Environment

The system is implemented by using Java programming language. Much of the application logic is implemented in a conventional object-oriented language Java. Java Server Pages (JSP) technology, which enables to mix regular, static HTML with dynamically generated content, is used to handle the presentation of the user interface.

Oracle JDeveloper 10.1.3 is selected as the development environment for the system. Oracle JDeveloper provides a single, visual, developer-friendly Integrated Development Environment (IDE) for Java programming. It is Java-based and runs on multiple platforms including Windows, Linux, Mac, and various Unix-based systems. It offers a rich set of features for designing, modelling, developing, testing, debugging, and deploying Java 2 Platform, Enterprise Edition (J2EE) applications.

The knowledge base of the system which consists of the ontology and the instance data has been processed, manipulated and queried by Jena API and presented to the user in HTML interface layout through a conventional Web browser. Jena is a Java-based open source API that is widely used in the development of Semantic Web applications. Since Jena is based on Java, it is easy to integrate it with any Java application. Jena provides a Java library for ontology languages and reasoning over data described in them. The ontology and instance data stored in files can be processed, manipulated and queried by Jena API. Jena supports RDF, RDFS DAML+OIL, and OWL, including a rule based engine and support for querying ontologies.

5. CONCLUSION

Semantic Web has been promoted as the future of the current Web. The Semantic Web aims at moving the current Web from being human understandable to being both human and machine understandable. The idea is to make Web content more accessible to machines by using semantic mark-up. Ontologies play a key role in the Semantic Web applications since they represent significant concepts and relationships for the application domain in a machine processable format. Ontologies abstract the knowledge of application domain and enable processing and sharing of this knowledge.

The Semantic Web is still under development. The W3C continues its work on development of specifications and standards. There has been intensive research activity in this field, and also applications based on Semantic Web technology are already beginning to appear. At this stage it is hard to judge to what degree this vision will eventually be implemented or when its use and deployment will become widespread. It is believed that in the near future the technologies being developed will make more efficient use of the Web and will offer more active, automated and intelligent services in all kinds of application areas, such as e-commerce, web services, e-learning, etc. Based on more machine-processable content, agents can browse, search, and match the information on the Web and can act on behalf of humans. Furthermore, these agents can also be implemented on other types of computing devices, such as mobile phones, PDAs, household appliances, etc. There remains significant opportunity in the area of the Semantic Web.

In this study, an example prototype Semantic Web based system for an academic department of a university has been designed and implemented. The system implemented in this study uses state of the art principles of the Semantic Web. An ontology, modelling the knowledge structure of the application domain has been developed and specified in a machine-processable format using Web Ontology Language (OWL). Protégé has been used as

the ontology editor. The system has been implemented using Java programming language. The knowledge base of the system which consists of the ontology and the instance data has been processed, manipulated and queried by Jena API and presented to the user in HTML interface layout through a conventional Web browser. The system and/or the ontology developed in this study can be taken as a model and can be easily adapted to other applications of the Semantic Web. The enhancement of the system requires further implementation. The system is an example research prototype. The scale of the system and amount of data being processed is small. Further optimization might be needed to enable the system to handle larger number of data and concurrent users. In addition, some applications, such as agents, web services, expert systems, and decision support systems can be implemented to benefit from the developed system.

REFERENCES

- Berners-Lee T., Hendler J., Lassila O. (2001): "The Semantic Web", Scientific American, May 2001.
- Bray T., Paoli J., Sperberg-McQueen C. M., Maler E., Yergeau F. (2006): "Extensible Markup Language (XML) 1.0 (Fourth Edition)", Retrieved April 20, 2007, from <http://www.w3.org/TR/xml/>.
- Brickley D., Guha R.V. (2004): "RDF Vocabulary Description Language 1.0: RDF Schema", Retrieved April 20, 2007, from <http://www.w3.org/TR/rdf-schema/>.
- Broekstra J., Klein M., Decker S., Fensel D., Horrocks I. (2000): "Adding Formal Semantics to The Web: Building on top of RDF Schema", Proceedings of the ECDL 2000 Workshop on the Semantic Web, Lisbon, Portugal.
- Fensel D., Horrocks I., van Harmelen F., Decker S., Erdmann M., Klein M. (2000): "OIL in a Nutshell", Proceedings of the 12th European Workshop on Knowledge Acquisition, Modeling, and Management (EKAW 2000), Lecture Notes In Computer Science, Vol. 1937, Springer-Verlag, 1-16.
- Fensel D., van Harmelen F., Horrocks I., McGuinness D., Patel-Schneider P.F. (2001): "OIL: An Ontology Infrastructure for The Semantic Web", IEEE Intelligent Systems, 16 (2), 38-45.
- Gomez-Perez A. (2002): "A Survey On Ontology Tools", OntoWeb Deliverable D1.3. Retrieved April 20, 2007, from <http://citeseer.ist.psu.edu/525623.html>.
- Gomez-Perez A., Corcho O. (2002): "Ontology Languages for The Semantic Web", IEEE Intellegent Systems, 17 (1), 54-60.
- Gomez-Perez A., Fernandez-Lopez M., Corcho O. (2004): "Ontological Engineering: With Examples from the Areas of Knowledge Management, E-commerce and the Semantic Web", Springer.
- Gruber T.R. (1993): "A Translation Approach to Portable Ontology Specifications", Knowledge Acquisition, 5 (2), 199-220.
- Guarino N., Giaretta P. (1995): "Ontologies and Knowledge Bases: Towards a Terminological Clarification", In N. J. I. Mars, (Ed.). "Towards very large knowledge bases: Knowledge Building and Knowledge Sharing" (25-32): Amsterdam: IOS Press.
- Horrocks I. (2002): "DAML+OIL: A Description Logic For The Semantic Web". Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, 25 (1), 4-9.
- Manola F., Miller E. (2004): "RDF Primer", Retrieved April 20, 2007, from <http://www.w3.org/TR/rdf-primer/>.

- McGuinness D.L., van Harmelen F. (2004): “OWL Web Ontology Language Overview”, Retrieved April 20, 2007, from <http://www.w3.org/TR/owl-features/>.
- Neches R., Fikes R., Finin T., Gruber T., Patil R., Senator T., Swartout W. (1991): “Enabling Technology for Knowledge Sharing”, AI Magazine, 12 (3), 36-56.
- Noy N.F., McGuinness D.L. (2001): “Ontology Development 101: A Guide To Creating Your First Ontology”, Stanford Knowledge Systems Laboratory Technical Report KSL-1-5 and Stanford Medical Informatics Technical Report SMI-2001-0880.
- Studer R., Benjamins V.R., Fensel D. (1998): “Knowledge Engineering: Principles and methods”. IEEE Transactions on Data and Knowledge Engineering, 25 (1-2), 161-197.
- W3C (The World Wide Web Consortium) (2003): “Semantic Web Status and Direction”, ISWC2003 keynote. Retrieved April 20, 2007, from <http://www.w3.org/2003/Talks/1023-iswc-tbl/slide26-0.html>.