KEMI-TORNIO UNIVERSITY OF APPLIED SCIENCES

Semantic Web Portals: Design and Development Technologies and Tools

Ansari Aftab

Bachelor's thesis of Degree Programme in Business Information Technology
Bachelor of Business Administration

TORNIO 2012

ABSTRACT

Ansari, Aftab 2012. Semantic Web Portals: Design and Development Technologies and Tools: Bachelor's Thesis. Kemi-Tornio University of Applied Sciences. Business and Culture. Pages 67.

Semantic Web is one important and relevant research area in computer science. A growing research attention to this field can be explained by the opportunities the Semantic Web could provide by representing and reasoning about semantic information. The objective of this thesis is to study the technologies for building information architecture and tools for technical implementation of Semantic Web Portals (SWPs). In particular, this thesis focuses on the concepts of ontology and the formalisms for representing ontologies within SWPs. These formalisms include, for instance, RDF (Resource Description Framework) and OWL (Web Ontology Language). There are many software tools, for example, Protégé, Jena and OntoStudio, intended for the practical implementation of ontologies and these are overviewed and analyzed in this work. In addition, the thesis focuses on two implementations of SWPs in Finland, namely, HealthFinland and MuseumFinland.

Based on these research results this thesis proposes the idea of building a SWP for Finnish educational system. The discussion of the information architecture for the proposed SWP is limited to Kemi-Tornio UAS due to the time allocated for thesis research. Nevertheless, the results of the work prove that this idea is relevant and implementable and will bring benefits to a wide category of users both general public and the education professionals.

The main research methodology used in this thesis is exploratory research based on the literature use and analysis. There are plenty of books and academic articles published in this research area. This research work is largely theoretical although the practical idea following the research results is presented and its implementation is suggested. Finally, opportunities for continuing this research work are discussed.

Keywords: Semantic Web Portals, Ontology, RDF (Resource Description Framework), OWL (Web Ontology Language), EducationFinland.

CONTENTS

ABSTRACT

FIGURES

FIGURES

1 INTRODUCTION

The information on the Web is growing remarkably today. Over the past two decades, Web pages as well as Web users seem to have increased enormously. The Web which had 1000 users in 1990 has now more than 1 billion users (Lausen & Stollberg & Hernandez & Ding & Han & Fensel 2004, 1). This rapid growth of Web users and the vast amount of information placed over the Web have also inspired the technological advancement in the world of Web. Lausen et al. (2004) point out two major success elements for this remarkable growth: the simplicity of publishing information on the Web and the access to new documents for software developers, information providers, and users. However, the simplicity which played a dominant role for the expansion of the Web has also brought some serious limitations that are hampering further development. The three major limitations discussed by Lausen et al. (2004) are the following: searches are imprecise, often yielding matches to many thousands of hits, and users face the task of reading the documents retrieved to extract the information desired. To overcome such limitations, the concept of Semantic Web (SW) has been proposed.

1.1 Semantic Web

The SW can be defined as “a Web of data that can be processed directly and indirectly by machines” (Berners-Lee 2001 as cited by Alesso & Smith & Craig 2004). The SW is an extension of the World Wide Web that holds machine accessible meanings of the containing data (Parsia & Patel-Schneider 2004). Studies explore that several tools and techniques have been developed over the past decade for the implementation of Semantic Web. However, the proper and universal implementation is yet to be achieved. As depicted from Alesso et al. (2004), the Semantic Web is under conceptual development process. The framework and the language used in developing this Web technology are proposed by World Wide Web Consortium (W3C). The SW developed with the help of the Resource Description Framework (RDF) and Web Ontology Language (OWL). Miltiadis et al. (2005, 90) emphasizes that the key enabler of the Semantic Web is the need of many communities to put machine understandable data on the Web which can be shared and processed by automated tools as well as by people.

Machines should not just be able to display data, but rather be able to use the data to automate, integrate and reuse across various applications.

Today, Web does not have any reliable way to process the semantics of the data as most of the Web contents are intended for human to read (Berners-Lee & Hendler & Lassila 2001). Data are published in the Hyper Text Markup Language (HTML) format on the Web pages. These data published on the HTML Web pages lack global schema. Therefore, it is difficult to use these data in a large scale and get them processed by machines. The HTML can format the data on the Web pages in a human readable form but it cannot process the meaning of the data. (Lausen et al. 2004.) To overcome this problem, Tamma (2010, 84) proposes that Semantic Web can represent information in a machine readable form and can maintain the human friendly HTML form at the same time. As discussed by Miltiadis et al. (2005), Extensible Markup Language (XML) enables the exchange of data across the Web but it does not process the meaning of that data. The use of Semantic Web will help process the meaning of data. The Semantic Web will bring about the structure to the meaningful content of Web pages, where software agents moving from page to page can readily carry out automated tasks. The objective of the Semantic Web is to provide a language that expresses both data and the rules of reasoning as a Web based knowledge representation. The Semantic Web layer cake depicted in Figure 1 adopted from (Berners-Lee 2001 as cited in Sure & Studer 2005, 195) shows the layering of the current state-of-the-art and future planned standards of Semantic Web. On the right side can be seen the current status of each layer.

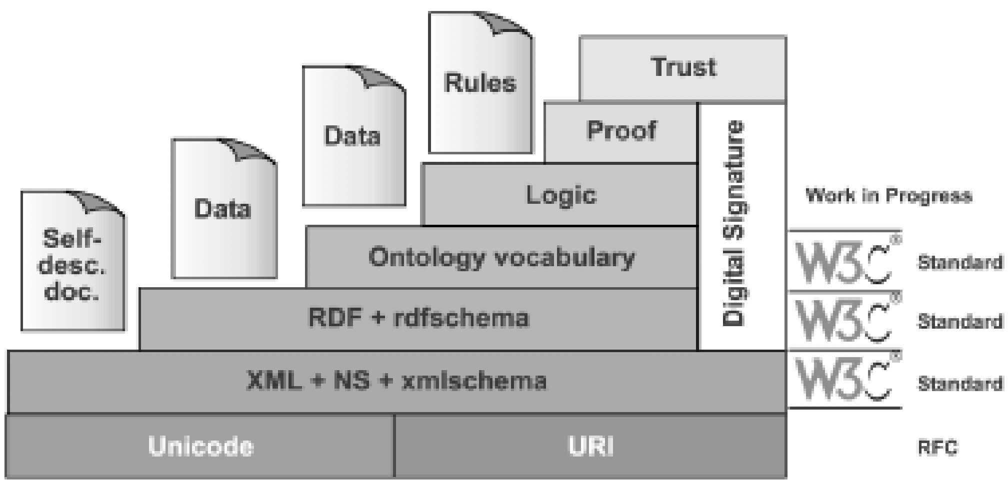


Figure 1. The Semantic Web Layer Cake (Sure & Studer 2005, 195)

Since the Semantic Web deals with processing the meaning of the data within Web pages, it is called Semantic Web (Miltiadis et al. 2005). Semantic Web applications are oriented towards human users, machines, and software agents. Semantic Web applications function with structured formal statements. These applications use a formal descriptive logic with links between data. The ordinary Web applications are primarily oriented towards human users. The ordinary Web applications operate using unstructured data with informal logic and links between documents. (Rovan & Jagust & Baranovic 2011, 245.) In the context of Semantic Web, the term Web 3.0 also seems to be discussed very frequently.

This term is used to describe various evolutions of Web usage and interaction together with several paths such as transforming the Web into a database, making content accessible by multiple non-browser applications, leveraging the technology of Artificial Intelligence, the Semantic Web, and the Geospatial Web. Berners-Lee et al. (2001) describes Semantic Web as a component of Web 3.0 which will offer an access to a greater scale of date. Since both the Semantic Web and Web 3.0 are under conceptual development process, people have different opinion about them (Ganz 2008, 34). Wolfram (as cited by Kobie 2010), argues that Web 3.0 is where "the computer is generating new information", rather than humans. According to Nova Spivack, the Chief Executive Officer (CEO) of Radar Networks (as cited by Nakate 2010), "Web 3.0 is a set of standards that turns the Web into one big database". Apart from these different opinions, Web 3.0 is also dependent on Artificial Intelligence. The Web 3.0 is a combination of Artificial Intelligence and Semantic Web which can group information in a manner that both computers and humans are able to comprehend. Since the concept of Semantic Web will play a central role in the evolution of Web 3.0, this term has become synonymous with Web 3.0.

1.2 Motivation

Semantic Web, being an emerging Web technology, offers different research directions for further development. There are different Semantic Web applications upon which research work can be done. As discussed by Rovan et al. (2011), some of the major research directions in Semantic Web are: Semantic Web Portals (SWPs), Social Semantic Web, Ontologies and Semantic Web, Human-computer interaction within Semantic Web, and Semantic Web Services. These research directions specialize in different areas of Semantic Web. However, the basic architecture remains the same.

SWPs study the semantic representation of data on the Web Portals. The SWPs add semantics to the contents and services on the Portal which eventually empower the Portal in comparison with normal Portals (Lei & Lopez & Motta 2012). Social Semantic Web deals with two major concepts which are Social Networking and Semantic Web (Mäkeläinen 2005). In other words, it studies the role and the use of Semantic Web technologies in social networks or social communities. The social Semantic Web also studies the representational mechanism and social aspects of data in social networks. Ontologies and Semantic Web to some extent, is a theoretical direction which studies the representation mechanism of data, OWL and ontology construction (Maedche 2011). Human-Computer Interaction within Semantic Web basically studies usability. The uses of natural language in making queries with systems and the ways the systems can accommodate those queries are also studied in this research direction (Garcia & Gil 2012). Roman et al. (2011) discusses Semantic Web Services as one of the recent research directions that studies the construction of Web Services based on Semantic Web technologies.

Even though all the aforesaid research directions are important and relevant, SWPs seem to be the most crucial research direction currently. HealthFinland and MuseumFinland are the two major research works about SWPs conducted in Finland. These research works also emphasize the significance and relevance of research work in SWPs. (Hyvönen et al. 2005.) Thus, SWP is chosen to be the core area of research for this thesis work.

1.3 Objectives

The objectives of this thesis are to study the major technologies used for building information architecture of SWPs, and to analyze the tools which are widely used for the technical implementation of SWPs. To accomplish the objectives, the major technologies of SWPs such as RDF/S, OWL, and SWSs are studied in detail. In addition, this work also focuses on the concepts of ontology and the implementation of ontology in SWPs. RDF/S and OWL are studied as formalisms for representing ontology in SWPs. Different software tools used for the practical implementation of ontology are analyzed. These software tools include Protégé, Jena, OntoStudio, D2RQ, and Sesame. As a practical output, an ontology representing Finnish higher education system is created using Protégé. Finally, this thesis proposes the idea of building SWP based on the research results of different tools, technologies and examples of SWPs studied in this work.

1.4 Structure of thesis

This thesis is divided into 9 different chapters. The scope of this thesis, the proposed research questions, and research methodology are discussed in chapter 2. In addition, this chapter presents a roadmap for this thesis including relevant topics. Knowledge representation and semantics, and ontology are the main focus of discussion for chapter 3. Semantic Web technologies such as RDF, OWL, and Semantic Web Services are studied in detail in chapter 4. Chapter 5 analyzes the existing Semantic Web tools such as Protégé, Jena, OntoStudio, D2RQ, and Sesame. Chapter 6 is dedicated for studying two major Semantic Web Portals in Finland namely HealthFinland and MUSUEMFINLAND in detail. An example of ontology representing Finnish higher education system, i.e. KTUAS Ontology is presented in chapter 7. Chapter 8 gives a brief and conceptual description of the proposed Semantic Web Portal, EducationFinland. Finally, chapter 9 includes the discussion and conclusion of this thesis work. In addition, it provides suggestions for further research that can be carried out to implement fully functional SWPs for Finnish higher education system.

2 RESEARCH QUESTIONS AND METHODOLOGY

This chapter defines the objectives of this thesis work, research questions, and the research methodology. The scope of the work is defined consisting of five related topics: knowledge representation, SW technologies, SW tools, examples of SWPs, and EducationFinland. These topics constitute a roadmap this thesis is built around.

2.1 Scope of the work

The objective of this work is to study the technologies and tools for designing and implementing SWPs. As a practical outcome, this work aims to present a conceptual description of how Semantic Web Portal for Finnish higher education can be built and how it will function, and an ontology design for the SWP for Finnish higher education system, called EducationFinland. To achieve these goals, the key aspects of the Semantic Web Portals are studied. These aspects include the technologies and tools used to build Semantic Web Portals, the role of ontology based semantics, and examples of existing Semantic Web Portals. During the study of the technologies and tools for designing and implementing SWPs, firstly the concept of semantics in Web technology is studied. Secondly, the role of ontology is explored. Thirdly, some of the key technologies used for building the SWPs are studied. These include Resource Description Framework/ Schema (RDF/S), Web Ontology language (OWL), and Semantic Web Services (SWSs).

In particular, this work focuses on the main aspects of how data can be published, searched, accessed, extracted, interpreted and processed semantically on the Semantic Web Portals. The work includes a literature review, an analysis of different research projects conducted in this field. The work also discovers possible directions for further research. The two major related projects studied in this thesis are HealthFinland at http://www.thl.fi/en_US/web/en, and MuseumFinland at http://www.museosuomi.fi/. This work also includes the research in tools for building Semantic Web Portals. The outcomes of this work are: a conceptual description of how Semantic Web Portal for Finnish higher education can be built and how it will function, and an ontology design for a Semantic Web Portal for Finnish higher education system.

To do this comprehensive study about Semantic Web Portals, several aspects of Semantic Web technologies are studied, analyzed, and tested. This thesis work studies the topics presented in Figure 2 to sequentially study and analyze different aspects of Semantic Web Portals.

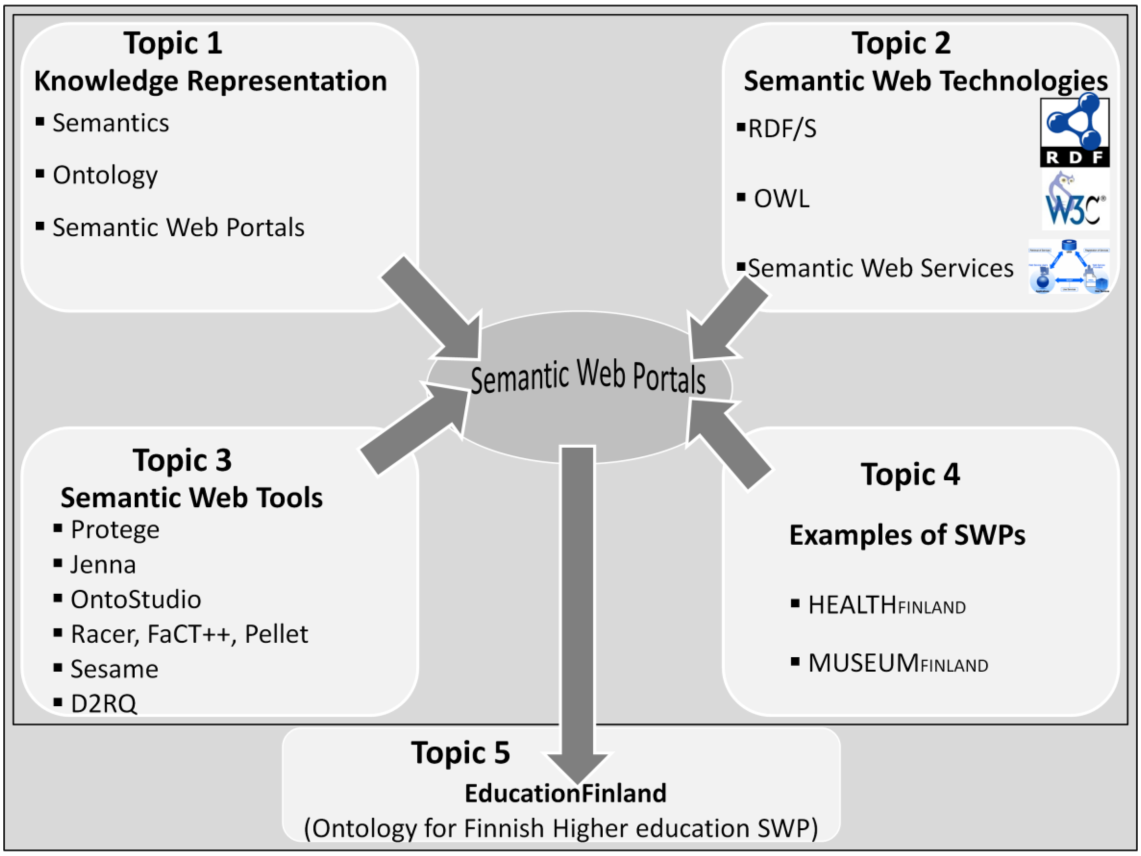


Figure 2. A roadmap for the thesis work defining the relevant topics

Figure 2 includes five topics which are the most important parts of this thesis. First, the concept "Semantics" is studied. The semantics is a part of knowledge representation. The semantic aspect is studied based on ontology. Following these studies, the technologies used for building Semantic Web Portals are studied as the second important aspect. This thesis explores some prominent technologies such as RDF/S, OWL, and SWSs. The concepts, basic syntaxes, and some sample codes of such technologies are presented and analyzed.

This work discusses two existing Semantic Web Portals to point out the applicability of the Semantic Web Portals, and the need for such a Portal for Finnish higher education system. The two selected Semantic Web Portals are HealthFinland and MuseumFinland. Further, the roadmap includes some of the widely used tools for building Semantic Web Portals. The features and functionalities, and the licensing of tools such as Protégé, Jena, OntoStudio, Racer, FaCT++, Pellet, Sesame, and D2RQ are tested. Eventually, based on the studies about the aspects of the Semantic Web technologies, an ontology design for EducationFinland is proposed as the practical outcome of this thesis work.

2.2 Research questions

Based on the objectives and the scope of this work, the following research questions are defined.

1. What is a SWP and what are the advantages and challenges in building and using it?

To answer this question, the definition and concepts of Semantic Web Portals are discussed. The connection between semantics and Web technologies is pointed out. In addition, the limitations in implementing semantic Web technology universally are analyzed. Topic 1 as illustrated in Figure 2 addressed in chapter 3 covers the answers to this research question.

2. What is the general information architecture of Semantic Web Portals and which technologies can be used to represent semantic information?

The information architecture of Semantic Web Portals is discussed to answer the first question here. The Semantic Web layer cake proposed by Berners-Lee et al. (2001) is discussed with particular focus on the ontology, RDF/S, and Web ontology language. The possibilities of different alternatives which can be used in specific hierarchy of the Semantic Web Layer cake are outlined. Based on the discussions about the different components of the hierarchal Semantic Web layer cake, the overall information architecture of Semantic Web Portals is presented. Some of the major existing Semantic

Web technologies are discussed. The ontology modeling tools are also discussed with some examples. Topic 2 detailed in Figure 2 and addressed in chapter 4 provides the answers to this question.

3. Which programming tools could be used to design and implement Semantic Web Portals?

Some of the widely used programming tools are studied and tested. Based on the experience with these software tools, their features and functionalities, and their licensing conditions, a few reasonable software tools are proposed. Topic 3 illustrated in Figure 2 and addressed in chapter 4 provides the answers to this question.

4. Would it be possible and reasonable to design and implement a Semantic Web Portal representing information about Finnish higher education system?

Topics 4 and 5 are addressed by this research question. The possibilities of implementing Semantic Web Portal for Finnish higher education system are discussed. Additionally, some of the advantages of implementing such a Portal are presented with some explanations and justifications. Two selected Semantic Web Portals HealthFinland and MuseumFinland are studied in chapter 3 as examples to support the possibilities of creating Semantic Web Portal for Finnish higher education system. Moreover, an ontology design plan is proposed for the Finnish higher education system in chapter 7. In addition, a brief description about the information architecture, user interface, and the role of KTUAS Ontology in building EducationFinland is presented in chapter 8.

2.3 Research methodology

The research methodology is exploratory research based on literature analysis. This research method is appropriate for this research as it will help to identify the opportunities in the field of Semantic Web Portal for Finnish education system. According to Pannereerselvam (2004, 6), exploratory research is an initial research which is conducted through a general study. This research method discovers the basis for general findings which are later explored by researchers and practitioners.

Therefore, this research method helps to build foundations of different hypotheses of research problems. The definition elements and the objectives of the exploratory method as described by Pannereerselvam (2004, 6) are appropriate for this work.

Although several research works have been conducted about Semantic Web Portals and many articles published, the research direction of Semantic Web Portals is still under development. Despite the fact that a lot of theoretical research works have been conducted, the practical aspects of Semantic Web Portals still seem to be limited and under development. In this thesis, relevant literature and research projects are analyzed.

Since the research about Semantic Web Portal is still mostly theoretical with further possibilities of practical applications, this research work relies basically on literature analysis and the availability of tools, technology, time or money required for the accomplishment of fully functional Semantic Web Portals. The required tools and technologies for building fully practical Semantic Web Portals are still under development. This unavailability of tools and technologies further emphasizes why exploratory research methodology is the most appropriate for this research. The direction for future research and techniques arises from the outcome of exploratory research methodology (Information Village 2009). Therefore, one of the purposes of using exploratory research in this work is also to develop further ideas for a conceptual description of how Semantic Web Portal for Finnish higher education can be built and how it will function, and an ontology design for the SWP for Finnish higher education system, called EducationFinland.

3 KNOWLEDGE REPRESENTATION

Knowledge Representation (hence forth KR), in the domain of Semantic Web, is known as the method of representing the semantics of data. KR uses a set of inference rules to achieve automated reasoning of data by the help of software agents. (Stroka 2005.) In the context of the SW, KR formalisms still have a room for development. Different approaches were proposed to overcome the limitations of KR formalisms in the SW. These approaches include the approach of metadata, RDF (Resource Dedcription Framework), and some ontology based KR language approaches. The metadata has 15 elements such as title, subject, description, source etcetera. The data on the Web is categorized and defined based on these 15 elements. This approach fails to achieve the goal of KR as it is impractical to describe all the data on Web based on only these 15 elements. The RDF has its own limitation as it cannot handle the negation. (Zarri 2002.) Despite these challenges, different approaches are tested to establish the relationship between the knowledge base and semantics. The goal is always to make the computer understand the meaning of the data it holds.

3.1 Knowledge representation and semantics

KR is an important part of Artificial Intelligence (AI). Hence, the concept of knowledge representation was brought into use several decades ago. Plato defined knowledge as justified true belief which was later negated by different epistemologists. They believed that knowledge could not be sufficiently defined based on true belief. (Sowa 2000.) In the field of information science, the knowledge representation seems to be quite prioritized as there are plenty of publications available on this topic. Knowledge representation today is mostly used as an approach for handling some key problems of information society such as structuring and storing information, searching and retrieving that information precisely and effectively (Weller & Katrin 2010). Therefore, the knowledge representation approach is significantly applied in information retrieval process. Such information retrieval process involves document indexing and development models of knowledge representation. Figure 3 shows the overall value chain of knowledge representation.

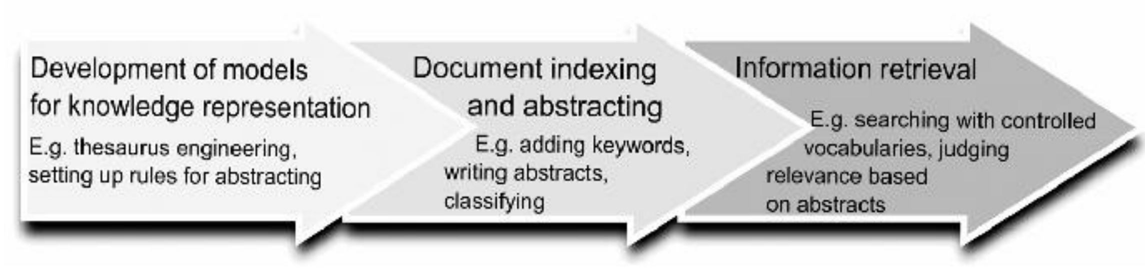


Figure 3. Value chain of KR, document indexing and information retrieval (Weller & Katrin 2010, 18)

The development of models for knowledge representation serves for setting up the rules for abstraction. According to Weller and Katrin (2010), indexing and abstracting are the two ways that can be used to describe a document's content. As for example, writing abstract of a document helps summarize the main contents of the document. Indexing mainly involves the task of adding content descriptive keywords. In Semantic Web, the knowledge is represented through ontology.

Semantics is one of the key aspects of semantic Web. The purpose of implementing semantics is to trace the meaning of the information on the Web. In other words, the machine is able to understand and interpret the meaning of the information correctly with the help of semantics. In the context of semantic Web, the semantics is meant to assist software agents in having a successful conversation by understanding the meaning of the contents on the Web. (Sheth & Ramakrishnan & Thomas 2005, 4.)

The core meaning of "semantics" is meaning itself. The use of semantics on the Web enhances the interactivity among Web agents. The Web agents can understand the meaning of the information being exchanged among them. Hence, the tasks they perform become semantically accurate. The application of semantics makes the Web contents machine usable contents which means the machine knows what to do with the Web contents as soon as it encounters them. The semantics can be implicit, informal or formal and can be hardwired into the Web by a human. The semantics can be hardwired in the very same way as the semantics of some symbols such as "+", "-", and "*" are hardwired into a procedure beforehand. The implicit semantics conveys the meaning based on a shared understanding derived from human consensus. The tags such as name, student number, address used in XML files do not define the meaning of themselves anywhere in the file rather these tags are interpreted by human consensus.

The informal semantics conveys the meaning through text descriptions. The meaning of tags defined in HTML can be taken as example for informal semantics. Formal semantics can be either for human processing or for machine processing. The formal semantics for human processing conveys the meaning through formal documentation of formal specifications of meaning. As for example, different axioms and definitions specified while creating ontology. The formal semantics for machine processing conveys the meaning by directly processing the contents using automated inference. (Uschold 2011.)

3.2 Ontology

Ontology, in the domain of philosophy, refers to the effort of describing any object which has just been discovered and has not got any description yet. The description made through ontology helps to place the very object with an identity among all the other existing objects. Philosophers since Aristotle have shown their great concern about knowing what exists and how to describe it. Ontology provides precise vocabulary which can be used to represent knowledge. This vocabulary specifies different entities, groups them, and connects them by defining relationships. The complexity of ontology including categories and relationships can reduce the potential understanding of how data can be used. Therefore, the ontology should be as simple as possible in order to avoid complications, confusions, and difficulties to use, maintain, and extend it. (Segaran & Evan & Tayler 2009.)

Ontology can be expressed with the help of Resource Description Framework (RDF) since the RDF is a Meta model. The formal rules of inference set by ontologies allow the software working on it to collect all the necessary information and draw the similar conclusions as human beings. (Colomb 2007, 99.) Semantic models describe groups of entities by defining different classes as object oriented models. The semantic models focus on the relationship between entities. Therefore, semantic models are more property-oriented than object-oriented. The entities in semantic models are members of a class based on the property they posses. The property in the semantic models is defined independently. This property definition can indicate the types of resources which can have property and the values the property can take on. (Sagaran et al. 2009.)

As for example, Figure 4 describes the domain and range of a property expressing eye colour.

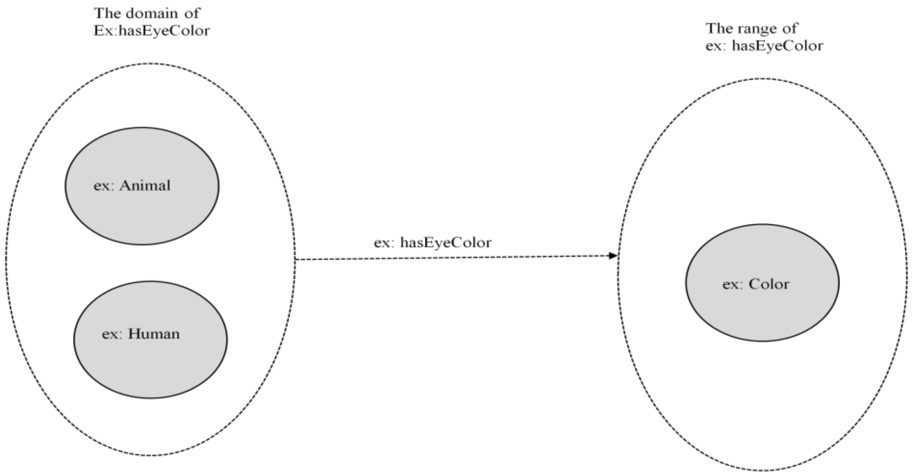


Figure 4. The domain and range of a property expressing eye colour (Sagaran et al. 2009, 130)

The property should indicate its domain to infer about the resources the property describes. When a property defines a type as its domain, anything defined by that property is of the same domain type. Similarly, if the property defines multiple types as its domain, all the resources described by that property are of the same domain types. The property should also specify a range to infer the value of the property. The value of the property can be inferred to be of single or multiple types based on the specification made by the property. There are differences between semantic and object oriented approaches to class and property definition. Suppose there is a small model of a person which provides information about eye color. In object oriented approach, an Animal class with a Human subclass is defined representing eye color through a variable. When constructing an instance of the Human class to represent Peter, the instance would have a variable to hold his eye color which can be set to “blue”. Now since this is an object oriented approach, Peter is both an animal and a human because the object representing him was constructed from the Human class. (Sagaran et al. 2009.)

Assuming example.org domain as a fictional semantic model which will give the prefix ex, it can be declared that the property hasEyeColor has a domain of Animal and Human with a range of Color.

ex:hasEyeColor rdf:type rdf:Property

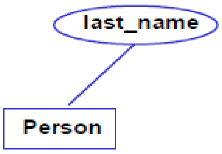
ex:hasEyeColor rdf:domain ex:Animal

ex:hasEyeColor	rdf:domain	ex:Human
ex:hasEyeColor	rdf:range	ex:Color

And the following statement expresses that Peter has blue eyes:
<http:semprog.com/people/Peter> ex:hasEyeColor
<http://rdf.freebase.com/ns/en.blue>.

The statement and definition of hasEyeColor given above infers that Peter is an animal and a human and that blue is a color, without having further assertions made about either Peter or blue. Since Semantic classes are defined based on properties, classes can also be defined in terms of the value of a property. For example, a class consisting of people with blue eyes can be defined. Therefore, in the case mentioned above, Peter does not necessarily need to be asserted to be a member of the class as his membership can be inferred with the assertion that he has blue eyes. (Mahalingam & Michael 1998.)

Ontologies are ultimately stored in pure textual format of some form or another. There are different standards available for storing ontologies in textual format. Some of the most common standards are KIF, Lisp, Clips, Loom, Ontolingua, and LDL formats. KIF (Knowledge Interchange Format) seems to be the most widely used standard for storing ontologies in textual format. The examples from Figure 5 explain how three different formats KIF, LOOM, and LDL represent the concept Person and a property called last_name. (Mahalingam & Michael 1998.)




```
KIF
(defrelation Person
        (subclass-of Person Agent)
        (class Person))
(deffunction person.last_name
        (function    person.last_name)
        (domain person.last_name Person)
                     (range person.last_name
              Name)
        (arity person.last_name 2))
```

```
LOOM
(loom:defconcept Person
:context Agent
:is-primitive (:and Agent)
:annotations
((documentation
"A person represents a human")))

(loom:
:context Agent
:is-primitive loom:binary-tuple
:domain Person
:range Name :attributes
       (:single-valued)
:arity 2
:annotations
((documentation
"The last name of Person.")))
```

```
LDL++
frame('Agent_Person'),
name('Agent_Person', 'Person'),
member('Agent_Person', 'Agent'),

slot('Agent_Person_last_name'),
name('Agent_Person_last_name',
     'last_name'),
defrelation person.last_name
domain('Agent_Person_last_name',
       'Agent_Person'),
range('Agent_Person_last_name',
       string),
```

Figure 5. Ontology representation in textual format (Mahalingam & Michael 1998, 3)

Ontologies represented in textual format are easier to port from one operating system to another as they do not store any platform dependent information in their representations. This feature of porting ontologies is very useful for knowledge sharing specially in a distributed and heterogeneous environment. Despite the fact that representing ontologies in textual format has these above mentioned great advantages, the textual representation of ontologies also has some disadvantages. One disadvantage of textual representation of ontologies is that such ontologies are difficult to comprehend due to the nature of how the information is presented to users. Therefore, if the information is not easy enough to understand, the use of the information will be very limited. Some of the problems of textual representation of ontologies can be eliminated through proper graphical representations of ontologies in combination with textual construct. (Mahalingam & Michael 1998.)

Graphical representations of ontologies are easier to understand in comparison with textual representation of ontologies. Even though the graphical representations are easier to understand, they are harder to port from one operating system to another because system-dependent information often must be stored in addition to the actual knowledge base. Today, system-independent language such as Java is used to avoid the need of storing additional information. (Mahalingam & Michael 1998.)

Entity-Relationship (ER) diagram is the most preferred and popular graphical representation method today to represent ontologies. Since there is a close relationship between the information spaces and databases, ER diagram with its simple format becomes more appropriate for representing ontologies. An ER diagram basically has nodes to represent entities or concepts, and arc to represent relationship between concepts. Graphically, entities are represented by rectangle, attributes are represented by ellipses, and relationships between entities are represented by diamonds. Figure 6 shows an example of ontology representation through ER diagram. (Mahalingam & Michael 1998.)

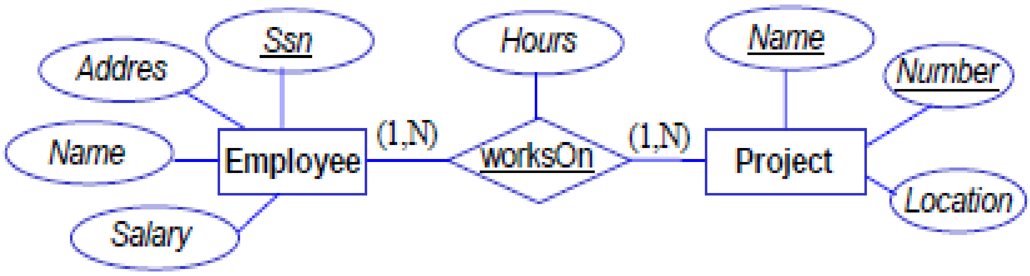


Figure 6. A simple ER diagram to represent ontology

Requirement analysis is the foremost and crucial step to be taken in order to proceed towards creating ontology. It seems very important to decide either to use semantic representation or a classical database in the beginning of the requirement analysis. Since the non-semantic approach already exists, there is no doubt that switching to the semantic approach would be more expensive and challenging. However, there are two major points which strongly support the idea of why an ontology based system is a better choice. First, it is easy to exchange and integrate the knowledge represented in semantic format with the knowledge from other source. Second, the implicit knowledge following from a semantic specification can be accessed by employing deduction algorithms. (Maedche & Staab 2011.)

Tool support is another very important aspect to be considered in terms of different knowledge representation options. There are several criteria which should be taken into account while reviewing the existing technologies. These criteria include the commitment to one specific tool or multiple tools, licenses associated with the available software/tools, the maturity of the software or the tools, the support offered by the tools vendor, and the compatibility of the tools. This becomes important while creating modelling formalism. For instance, RDF (S) can be a good choice in case large amount of data have to be handled and a less expressive formalism is sufficient. However, if more expressive means of knowledge representation is required with moderate size of represented information, OWL DL could be a better choice. (Hitzler & Krötzsch & Rudolph 2010.)

After the modelling formalism is agreed upon, the requirements of the ontology should be specified more explicitly based on what domain has to be modelled, the aspects of the domain to be covered, the level of detail, the tasks to be completed with the help of the ontology, and the types of expected inferences. These basic criteria help choose individuals, classes, roles, and the degree of axiomatization have to be chosen. Finally when comes the creating the ontology, there are two possible approaches for that purpose. First, the ontology can be created from scratch. In this case the classes, roles, and individuals have to be defined. The second approach is to combine the existing ontologies and form a new one. This approach requires inclusion, restriction, and refinement. (Moria 2007.)

Ontology specifies knowledge about some domain. Therefore, creating ontology can be seen as transferring knowledge into a computer- accessible form. In most cases, a domain expert is asked to formulate his knowledge in such a way that it can be written down in some KR formalism. A domain expert refers to a person who is knowledgeable in the domain of interest and is acquainted with the used ontology language. However, it is not always possible for a domain expert to formalize his knowledge in such a form which can be written down in some knowledge representation form. As for example, the domain expert sometimes might not be able to present his knowledge clearly. Therefore, more than one domain experts are interviewed in order to avoid such ambiguity. This communication process with more than one domain expert sometimes causes information loss and errors as there might arise misunderstanding. The introduction of redundancy, feedback and double check in the process of interview can overcome such danger of misunderstanding. (Hitzler et al. 2010.)

Natural language documents such as books and other textual resources such as magazines, Web pages, and journals can also be seen as the source of knowledge. These sources are more practical when large amount of directly accessible knowledge is sought. However, it is a tough task to extract formal specifications from arbitrary written texts. The other important step is to transform the linguistic structure into a logical description. This transformation is an intricate task which helps grasp semantics from a grammatical structure of sentences of a natural language. The semantic relationships between the involved words are the aspects that have to be considered while converting language into a formal representation. Apart from this, other existing ontology can also be considered as a source while creating a new ontology. (Lausen et al. 2007.)

In some cases, the source which has to be used for creating ontology already has some structure in terms of semantic interdependencies. As for example hyperlinks, wiki articles that contain reference of other related articles. These structured sources can be directly transformed into an RDF or OWL representation. File system is considered to be the semi structured source of information. There are also some other source of information which is directly accessible. Database is an example of such source of information. The content stored in a relational database is possible to be translated into RDF or OWL. This kind of translation requires some additional information about the source and structure of the content. Such additional information helps comprehend how a row of table in a database can be transformed into a set of RDF. Database schema is helpful in retrieving the additional information. (Hitzler et al. 2010.)

Ontology itself is a structured source of knowledge which can be reused to create a new ontology. It is a wise idea to look for some existing similar types of Ontologies and use them fully or partially to build a new ontology. In some cases more than one existing Ontologies have to be reconciled. This reconciliation sometimes becomes a tough task to manage as they might depend on different ways of modelling, naming schemes, and languages. In order to overcome the challenges arisen during the integration of different ontologies into one, ontology mappings are used. The ontology mapping clarifies how the individuals, classes, and the roles of one ontology correspond to those of others. There are different approaches for applying ontology such as manual specification of ontology, automatic determination of ontology. (Hitzler et al. 2010.)

A good ontology is the one that fulfil the intended purpose set in the beginning of creating ontology. There are several basic criteria which have to be satisfied by ontology. These criteria are logical criteria, structural and formal criteria, accuracy criteria. The logical criteria help check ontology characteristics based on purely logical levels. Ontology also requires a model which can characterize its domain. In absence of such model, the ontology would be inconsistent. Therefore, it is very important for a good ontology to have logical consistency. This would help avoid modelling errors. During ontology creation, the logical consistency should be checked frequently to keep the modelling on the right track. Coherence is another important aspect of ontology that has to be evaluated while observing the logical criteria. The existence of unsatisfying classes causes the ontology to become incoherent. Therefore, it is always a good idea to define a new class only if it has instances. (Aquin & Schlicht & Stuckenschmidt & Sabou 2011.)

Structural and formal criteria examine the nature of the used classes based on the subclass hierarchy of the ontology. The rigidity of the class is considered while evaluating its general qualities. If every member of a class cannot cease to be a member without losing existence, such class is considered to be a rigid class. If a member of a class can do so, then such class is treated as anti-rigid class. On the basis of the above mentioned criteria, classes can be defined as rigid, anti-rigid or none of both. These criteria examine if the ontology captures the desired aspects of the modelled domain accurately. It is not possible to check real-world-conformance of ontology entirely automatically. Therefore, it is a subjective question to know if the ontology represents the certain or desired aspects of reality. (Aquin et al. 2011.)

There is not any particular way of modelling a situation when creating ontology. This is why different approaches are used. However, there are some aspects that have to be considered as they might cause flaw in the ontology creation. As for example, disjoint statements are neglected in most cases. This omission of disjoint statements causes the loss of potentially useful consequences. It is also important to assign enough characteristics to roles as they enable several useful deductions. Too specific domain or ranges should not be chosen as ontology is not meant to cover every tiny detail. The use of quantifier should be done very carefully so that the intended meaning is correctly cast into role quantifications. In this case existential quantification can be used as default. However, it is also important to consider that the universal quantification alone cannot enforce the existence of a respective role. (Hitzler et al. 2010.)

4 SEMANTIC WEB TECHNOLOGIES

The semantic Web which is also known as an extension of the World Wide Web allows to draw conclusions by processing the meaning of the contents on the Web. Some of the major technologies of the Semantic Web are RDF, OWL, and SPARQL.

4.1 Resource Description Framework (RDF)

The Resource Description Framework is a formal language which represents structured information. The RDF enables and allows applications to exchange data on the Web without changing their original meaning. Unlike, HTML (Hypertext Markup Language) and XML (Extensible Markup Language), the RDF not only displays the documents correctly but also allows for further processing and recombination of the information the documents contain. Today, every programming language has libraries for reading and writing RDF documents. RDF stores are used for keeping and processing large amounts of data. These stores are also known as triple stores. In some specific application areas, for example RSS 1.0, RDF is also found to be used for exchanging metadata (Hitzler et al. 2010.) RDF can conceptualize any entities in the universe as a resource which can further be identified with a Universal Resource Identifier (URI). Since URIs can identify anything as a resource, the subject, the object and the predicates in RDF are always resources. These resources are called the URI reference. (Segaran et al. 2009.)

RDF is based on simple graph-oriented data schema which describes a direct graph, i.e. a set of nodes linked by direct edges or arrows can be represented by RDF graph. As can be seen from Figure 7, there are two nodes and an edge which have labels with identifiers which distinguish them.

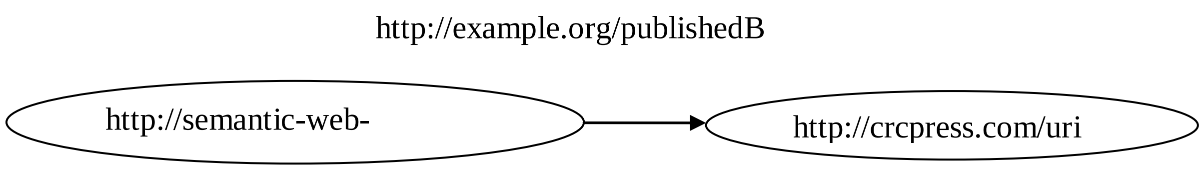


Figure 7. A RDF graph describing relationship between two objects (Hitzler et al. 2010, 20)

The graph for the representation of RDF makes the task of combining RDF data from multiple sources easy. Therefore, graph is preferred for representing RDF. The graph in Figure 7 expresses the relationship between two nodes. The first node contains name of book, for example foundations of Semantic Web, and the second node contains name of press. The arrow has been used to establish relationship between these nodes which says “publishedBy”. Therefore, the whole graph expresses that the book, Foundations of Semantic Web, was published by CRC press. Figure 7 also shows the use of URIs http://semantic-web-ok.org/uri, http://example.org/publishedBy, and http://crcpress.com/uri in order to distinguish these resources from others. The URIs are generalization of URLs, Universal Resource Locators. (Hitzler et al. 2010.) The URLs tell about where the specific information can be found and provide a unique identifier for the information. The URLs are subsets of URIs.

URIs are simple strings which have a scheme name followed by a colon, two slashes (://), and a scheme specific identifier. The scheme in the URIs is used to identify the protocol used. This protocol uniquely identifies the resource by using the scheme specific identifier. (Hebeler & Fisher & Blace & Perez-Lopez 2009, 480.) In most cases URIs uses either “http” or “https” scheme for identifying the resources. Even though every URL is a URI, all the URIs are not URLs (Tazzoli & Castagna & Campanini 2012). Therefore, typing the URIs identifier into a Web browser will not produce any information. The URIs can be shortened using namespace to the base URI during RDF representation. In this case, only the distinctive part of the identifier is mentioned. i.e. http://www.w3.org/1999/02/22-rdf-syntax-ns# and http://www.w3.org/1999/02/22-rdf-syntax-ns#type can be written as **rdf:type**. In case of not having identifier for the resource which is to be referenced, the RDF facilitates with “anonymous” or blank nodes. Blank nodes are used to represent the resources which have no other way to address with proper URIs. (Hitzler et al. 2010.)

The RDF is a syntax independent model that represents resources and their corresponding descriptions. The RDF model describes Web resources and objects which are uniquely identifiable by URIs. These resources are described using property names and values associated with resources. A collection of such properties referring to the same resource is known as a description. The three major components of RDF are Resource, Property, and Statement. (Selcuk & Huan, & Suvarna 2011.) These three key components have been described below briefly.

Everything which is described by RDF expressions is resource. In case of Web resources, any single object, Web page, part of Web page or the entire website can be considered as resources. Even an object which is not directly accessible through Web page, for example, any printed book, can also be considered as resource. Any attribute, aspect, characteristic or relation which is used to describe the resources is called property. The property carries a specific meaning that defines its permitted values. The property also bears meaning about what kind of resources it can describe and what relationship it has with other properties. When a specific resource is given attribute with some values associated with the attribute, this whole construction becomes statement. These three major components resource, attribute, and values are called subject, predicate, and object of the statement, respectively. (Selcuk et al. 2011.)

The RDF uses XML (Extended Markup Language) namespace mechanism to uniquely identify the property names. A namespace is a context or a setting that gives a specific meaning to what might otherwise be a general term. The XML namespace provides a method for removing ambiguity in identification of semantics and conventions. This method governs the particular use of property names by uniquely identifying the governing authority of the vocabulary. Therefore, RDF can define and exchange semantics among communities by using namespace. (Selcuk et al. 2011.) Figure 8 is a RDF description of a document that shows the use of namespace.

```
<?xml version = "1.0"?>
<rdf:RDF
xmlns:rdf="htpp://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:my="htpp://mymetadata.org/schema/">
<rdf:Description about="htpp://www.asu.edu/namespace/">
<my:Title>NamespaceFAQ</my:Title>
<my:Description>
This is the page of FAQ for ASU namespace.
</my:Description>
<my:Date>2001-06-14T09:46</my:Date>
</rdf:Description>
</rdf:RDF>
```

Figure 8. An RDF model for a document with use of namespace (Selcuk et al. 2011, 6)

Line 1 in Figure 8 defines the XML version used in the document. Lines 2-4 are dedicated for defining the root element rdf:RDF. The document has used two namespace prefixes rdf and my which are applicable to the RDF description in line 5-11. To give reference to the corresponding schemas, the URIs have been associated with the namespace declarations. The rdf:Description in line 5 is an element in the context of the rdf namespace which describes the resource to corresponding URI, http://www.asu.edu/namespace/. The URI has been further specified in attribute about. (Selcuk et al. 2011.)

RDF containers hold collections of resources. For example a list of actions performed by a certain group of people or an article written by a group of authors can be represented through RDF containers. There are three types of RDF container objects which can facilitate different groups of resources or literals. These RDF container objects are Bag, Sequence, and Alternative. Bag can be defined as an unordered list of resources or literals. Sequence represents the ordered list of resources or literals. Alternative holds a list of resources or literals which can represent alternatives for the single value of a property. (World Wide Web Consortium Recommendation 2004, 39-41.) Now based on the information that the BIT department of edu.tokem.fi offers courses: Usability, Databases, Competence Development, a simple bag container can be created. Figure 9 shows a simple RDF bag container. The container holds a list of courses offered by BIT department (edu.tokem.fi).

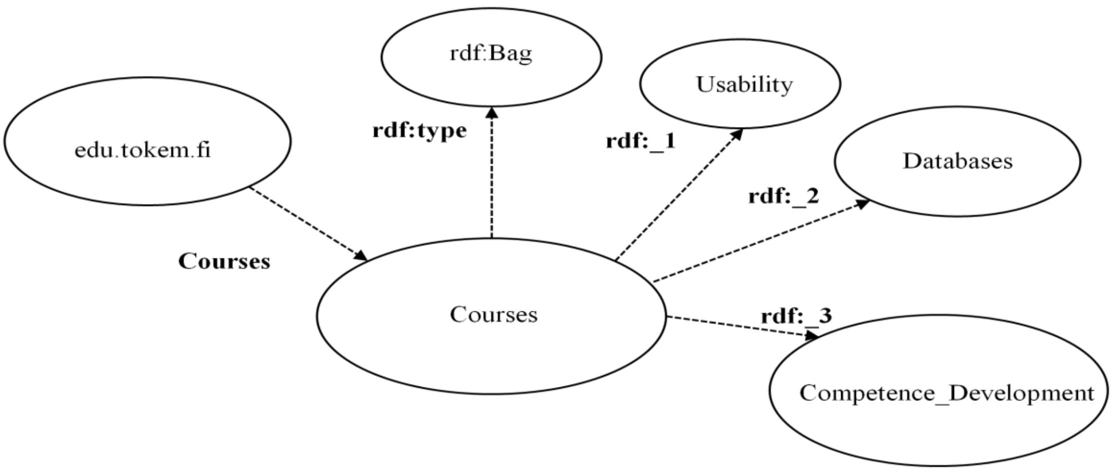


Figure 9. A graph model of simple Bag container example

The above shown graph can be converted into RDF/XML representation. The conversion of the above graph model of RDF bag container into RDF/XML representation is in Figure 10.

```
<rdf:RDF>
<rdf:Description about="http://edu.tokem.fi/courses">
<my:Courses>
<rdf:Bag>
<rdf:li resource="http://edu.tokem.fi/courses/usability">
<rdf:li resource="http://edu.tokem.fi/courses/databases">
<rdf:li resource="http://edu.tokem.fi/courses/competence_development">
</rdf:Bag>
</my:Courses>
</rdf:Description>
</rdf:RDF>
```

Figure 10. An RDF/XML model of simple Bag container example

4.2 Web ontology language (OWL)

Web ontology language (hence forth OWL) is an extension of RDFS (Resource Description Framework Schema) which has been built in order to define classes, properties, and the relationships among them with strong reasoning and inferences (Sagaran et al. 2009). The OWL is more expressive and scalable in comparison with the RDFS and is used to model different ontologies. Therefore, it can be apparently assumed that the OWL has taken the standard of conceptual modelling of ontology to next level than the RDFS. (Hitzler et al. 2009.) Since the OWL is built on the RDF and the RDF Schema, the RDF's XML syntax is still utilized in this case. The reason behind using the OWL and not the simple RDF/XML is that the syntax of the RDF/XML is not very readable. There are a few other approaches that OWL utilizes in order to make the syntax more readable. For example the OWL uses XML based syntax which are not based on the RDF conventions which in fact makes the syntax human readable. Sometimes abstract syntaxes are defined in the language specification documents to make the syntax more readable. Also the OWL uses the UML (Unified Modelling

Language)-based graphical syntax so that the OWL would become easy for people to get familiar with. (Antonio & Harmelen 2011.)

Referring to these approaches mentioned above, it can be vividly concluded that the OWL, with the help of RDF's XML syntax and other additional approaches used in it, can perform better reasoning. Additionally, the RDF is a data model for objects or resources which defines the relationship between these objects. The RDF being written in XML syntax can only hold a simple semantics for its data model. The RDFS adds some additional vocabularies that describe the properties and classes of the RDF resources. Now we infer from the above paragraph about the OWL that the OWL adds reasoning and inferences in the data models.

The introduction of XML serialization made it possible to create a data model for objects and relationships among them. The RDFS was developed in order to describe the properties and classes of the RDF resources. Though the RDFS is considered to be an ontology language which contains classes and properties, subclasses and super classes, and also utilises the concept of domain and range, it does have some limitations. The RDFS despite having all the features mentioned above suffers from its dependence on domain-specific and case specific details. This shows that the RDFS does not have enough expressiveness. The RDFS fails to establish some important relations between classes such as classes of equivalence, disjointedness, and cardinality and characteristics of properties. Above mentioned limitations of the RDFS are some of the major limitations that the RDFS suffer from. Though most of these issues have been mentioned in section 4.1, I pointed them out here to show how the evolution of the OWL is connected with the RDF/RDFS. I have briefly explained how the development of OWL was influenced by the development of the RDFS.

Since the RDFS lacked the required degree of expressiveness in the cases such as defining classes of equivalence and disjointedness, and cardinality and characteristics of properties, two alternative languages were developed concurrently to overcome the limitations of the RDFS. These languages were Ontology Inference Layer (OIL) and DARPA Agent Markup Language (DAML). The OIL was developed in Europe whereas the DAML was developed in the U.S. The commonality of both languages was that they were build on top of the RDFS. These two languages were submitted to the W3C for the standardization. This is when the W3C combined the two languages together and developed OWL. The OWL was also layered on top of the RDFS. The development process of the OWL has been presented in Figure 11 below. (Mostafa & Zhengbo 2011.)

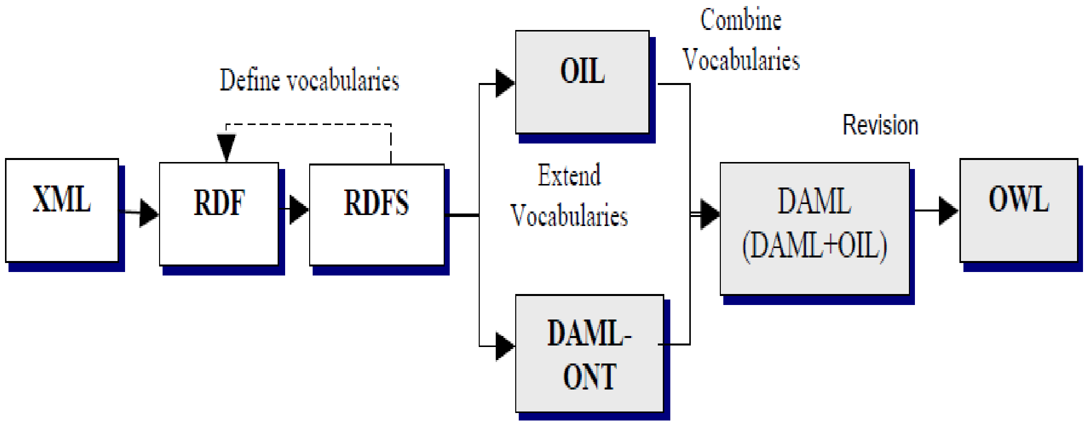


Figure 11. The development of OWL (Mostafa & Zhengbo 2011, 3)

The OWL was first recommended by the World Wide Consortium (W3C) in 2004. Today, the OWL, being the most popular language for creating ontology, has also become the largest recommendation of the W3C. As it has been explained that the OWL is built on RDF Schema, it is worth knowing what difference the OWL brings in creating ontology. The OWL is made up of RDFS and some additional constructs for expressiveness. This means the OWL works for the same purpose as RDF Schema does. The purposes are defining classes, properties and the relationships among them. However, the difference is that the OWL can define much more complex and richer relationships among the classes and properties which RDF Schema cannot do. This first version of OWL was called OWL1 which was later enhanced with some features and was versioned as OWL 2 in 2008. Therefore, the latest standard from W3C is the OWL 2. About the compatibility issues between these two standard OWL 1 and OWL 2, the good news is that the ontology created by using OWL 1 can also be recognized by the OWL 2. This implies that OWL 1 is now like a subset of OWL 2 and is fully compatible to OWL 2. (Liang 2007.)

The OWL, despite being rich in expressiveness, has some serious limitations in terms of treating data types. As for example the OWL does not support negation in data types. Additionally, since the OWL only allows a single value in a data type domain, some expressions such as greater than or equal to (≥), less than or equal to (≤), and so on cannot be assigned. The OWL also does not support user defined data types. Though the OWL uses the XML Schema which allows the user defined data types but such user defined data types cannot be used with the OWL as the OWL has no proper way to reference such user defined data types (Bruijn & Fensel & Polleres 2004.) It also seems that many potential users would not adopt the OWL due to these limitations as it is very crucial for the users to be able to define the data types and data type predicates. The W3C also seems to be quite concerned about such limitations and is working on addressing these issues. OWL –Full is found to be proposed as the extension of the OWL DL in order to overcome the limitations of the data types in OWL. (Pan & Horrocks 2005.)

4.2.1 The sub languages of the OWL

There are three different sub languages in OWL 1.0 which are OWL Lite, OWL DL, and OWL Full. These languages can be used by specific users and implementers based on their requirements. (Breitman & Casanove &Truszkowski 2007.) To understand the difference between these languages and how they fulfil different requirements, a brief introduction of each language has been presented below.

The OWL Lite serves better to those implementers and users who require a classification hierarchy and simple constraints. For instance, the OWL Lite supports only 0 and 1 as cardinality value when cardinality is applied as constraints. (World Wide Web Consortium 2004.) The OWL Lite does not hold enumerated classes, disjoint statements and arbitrary cardinality. Due to these limitations, the OWL Lite cannot allow the use of some class properties such as owl:oneOf, owl:unionOf, owl:complementOf, owl:hasValue, owl:disjointWith, and owl:DataRange. (Antoniou & Harmelen 2011.) Therefore, one of the differences between the OWL Lite and the OWL DL is that the former one has lower formal complexity which practically means less expressive constructs.

Despite the fact that the OWL lite is the least expressive language in comparison with the other species of OWL such as OWL DL this language do requires reasoning with equality. This reasoning equality eventually increases the computational complexities.

There are some other limitations of the OWL Lite too. As for example, we know that the functional properties can be defined with a cardinality of 1 at most when we apply cardinality as constraints. However, when two instances of such properties carry the same domain value but a different range value. In this case the semantics of the OWL Lite treat the two instances in this range to be equal. Therefore, deriving equality in OWL Lite is non-intuitive. (Bruijn et al. 2004.)

The OWL Full is suitable for the users who wants maximum expressiveness. The OWL Full uses more holistic approach in comparison of other species of the OWL. For example the OWL Full can treat the same class as a collection of individuals and as an individual in its own right. (World Wide Web Organization 2004.) The OWL Full uses all the other language's primitives. These primitives are combined in arbitrary ways with the RDF and RDF Schema. Since the OWL Full is fully compatible with the RDF and RDFS, any legal RDF document can be considered as a legal OWL Full document. Also any valid RDF/RDFS conclusion is also a valid OWL Full conclusion. (Antonio & Harmelen 2004.)

The OWL DL (Web Ontology Language Descriptive Logic) seems to be the most popular and widely investigated sub language of the OWL. This language is also seen as an alternate notation of the Description Logic Language SHOIN. (Bruijn & Polleres & Lara & Fensel 2005.) The OWL Full provides efficient reasoning support by restricting the application of the OWL's constructors between OWL and RDF. On the contrary the OWL DL also loses the compatibility with the RDF due to above mentioned restriction. As a result, an RDF document has to undergo some extensions and restrictions in order to be a legal OWL DL document. However, every legal OWL DL document can be considered as a legal RDF document. (Antonio & Harmelen 2004.)

Analyzing the explanations presented about the above mentioned sub languages of the OWL 1, we discover that all of the above mentioned sub languages have their own importance. The appropriateness of these languages is determined based on the users' requirement. We also noticed that the requirement in this case mostly deals with the degree of expressiveness, acceptance of the data types, and the support for the reasoning. Also the above mentioned sub languages belong to the OWL 1. The OWL 2 has its own sub languages.

According to the W3C working draft of 27th march 2009 published on the official website of W3C at http://www.w3.org/TR/2009/WD-owl2-overview-20090327/ , the OWL has been defined as an ontology language with formally defined meaning. It has also been stated that the OWL 2 ontologies have classes, properties, individuals, and data values which are stored as Semantic Web documents. Based on this information it becomes very apparent that the basic features of the OWL 2 are same as OWL 1. However, there are some differences that have been pointed out by the same working draft of W3C. The draft claims that the OWL 2 has some additional features which OWL 1 does not have. These new features include some new expressivity with keys, property chains, richer data types, data ranges, qualified cardinality restrictions, asymmetric, reflexive, and disjoint properties, and enhanced annotation capabilities.

The sublanguages of the OWL 2 are OWL 2 EL, OWL QL, and OWL 2 RL. These sub languages are also used based on the user requirements and the application scenarios. The OWL 2 EL is used when there is need of the polynomial time algorithms for the standard reasoning task in large ontologies with major concern over the expressivity. The OWL QL deals more with the relational databases. Therefore, this language is appropriate when a large number of individuals have to be organized by a lightweight ontology. In this case relational queries are used to access the data. The OWL 2 RL utilizes the rule-extended database technology which operates on the RDF triples. Therefore, this language is used for operating directly on data in the form of RDF triples. (World Wide Web Consortium 2009.)

4.2.2 OWL Syntax

Referring to the introduction chapter of the OWL mentioned above we know that the OWL is built upon the RDF/RDFS and that it uses the RDF’s XML syntax. Since the RDF/XML syntax is complex to read, some other syntactic forms are also used in OWL. There seems to be two different syntaxes which have been defined as standard syntaxes for the OWL. These two syntaxes are called OWL RDF syntax and OWL abstract syntax. As shown in different literature these two different syntaxes are used for different purposes in different situations. As for example the former one is mostly used for data exchange. The later one is only used with OWL DL. The OWL RDF seems to be widely used in comparison with the OWL abstract syntax. For the convenience in describing the syntaxes I would like to break down the structure of the OWL into some important parts such as header, classes, roles, and individuals. (Hitzler & Krötzsch & Rudolph 2010.)

4.2.3 Header

The header is an important part of the OWL document. Though the information presented in the header part of the document does not have a direct impact on the information expressed by the Owl ontology, it is still considered to be important for some other reasons. The header section of the OWL document contains mainly information about the namespaces, versioning, and annotations which are very similar type of information presented in the head section of normal HTML documents. Let me explain the contents of the header section of the OWL document with an example of a few syntaxes.

```
<rdf:RDF
xmlns:owl =http://edu.tokem.fi/2012/owl#
xmlns:owl =http://edu.tokem.fi/2012/rdf-schema#
```

Assuming the syntaxes mentioned above as a part of OWL document, the first line <rdf:RDF is a syntax that holds the root elements of the OWL documents. This root element also specifies the namespaces used in the documents. As for example the line 2 xmlns:owl =http://edu.tokem.fi/2012/owl# and the line 3 xmlns:owl =http://edu.tokem.fi/2012/rdf-schema# are the namespaces. Another important aspect of syntaxes used in the header section of OWL document is that the namespaces and other information such as comments, version control, and inclusion of other ontology can be grouped together. When grouping such information together owl:Ontology syntax is used to hold such group of syntax. The example codes below shows the grouping of namespaces, comments, version control, and inclusion of other ontology. (Antonio & Harmelen 2004.)

```
<owl:Ontology rdf:about="">
<rdfs:comment>sample ontology</rdfs:comment>
<owl:priorVersion
```

```
rdf:resource=" http://edu.tokem.fi/2012/priorVersion"/>
<owl:imports rdf:resource=" http://edu.tokem.fi/2012/second "/>
<rdfs:label>Second Ontology</rdfs:label></owl:Ontology>
```

A few lines of codes presented above outline a group element that contains comment, version control, and also the information about importing external ontology. Several literature works seem to be emphasizing on paying attention to the transitive property of owl:imports. Therefore, it is always wise to take into the account the transitive property of owl: imports while creating ontology. Let me explain what the transitive property of owl:imports exactly mean. The transitive property in this case means that if ontology A imports ontology B, and ontology B imports ontology C, then ontology A also imports ontology C.

4.2.4 Class

The class, like in RDF, is one of the major elements of OWL. The classes are represented with a specific syntax owl: Class. The OWL has two pre-defined classes which are owl:Thing and owl:Nothing. The owl:Thing is the class which is a general class and hence all the classes we create are created under the owl:Thing. The class owl:Nothing is an empty class. There are other additional properties of class that can be defined such as class id, class equivalence, and class disjointedness. It is also possible to create a sub class of an existing class. (World Wide Web Consortium 2004.) A few examples of such properties have been given below.

```
<owl:Class rdf:about="bitDepartment">
<owl:disjointWith rdf:resource="#bm"/>
<owl:disjointWith rdf:resource="# bitDepartment "/>
</owl:Class>
<owl:Class rdf:ID="facultyMember">
<owl:equivalentClass rdf:resource="#academicStaffMember"/>
</owl:Class>
```

4.2.5 Property elements

The property elements are very important in ontology development as they play a vital role in defining what kinds of objects and data the ontology holds. The OWL allows two different types of property elements which are object properties and data type properties. The object properties assist establishing relationship among various objects. As for example, an object property isWrittenBy can be defined in order to relate two objects Books and Authors. This object property can be represented through the OWL syntax in the following way <owl:ObjectPropertyrdf:ID="isWrittentBy"><owl:domain rdf:resource="#books"/><owl:range rdf:resource="#author"/></owl:ObjectProperty>. The data type properties define the data type and values that objects hold. As for example an object author can hold data type name which can be of type string. This example can be presented in OWL syntax as <owl:DatatypeProperty rdf:ID="name"><rdfs:range rdf:resource="http://www.w3.org/2001/XLMSchema #string"/></owl:DatatypeProperty>. As there is no predefined data types definition facilities in OWL, XML data types are used for this purpose. (World Wide Web Consortium 2004.)

Based on the richness of the OWL in terms of a wide range of functionalities and high expressiveness, the OWL seems to be quite appropriate for the use in practice. The specialty of OWL to better present the semantics of the knowledge in human readable form has increased its popularity. Moreover, the reasoning capacity supported by the OWL has added importance of the OWL in practical use. Most of the limitations of the RDF/RDFS such as the readability, lack of sufficient expressivity seem to be overcome with the use of OWL. As it depicts from several literature that the use of the OWL is getting wider and wider constantly, it would not be an unjust to conclude that the OWL has become an integral part of the semantic Web technology today.

4.3 Semantic Web Services (SWSs)

Web Services are interfaces to applications that can be published, located, and invoked across the web. Web Services can be identified by Uniform Resource Identifier (URI), and they can also be defined, described, and discovered by XML artifacts. In addition, Web Services can directly interact with other software applications with the help of

XML based messages via internet based protocols. (Dutta 2008, 48.) However, one limitation of such Web Services is that they cannot interpret or represent meaning of inputs and outputs or other similar constraints. Therefore, SWSs are proposed to overcome this limitation. With the use of Semantics, the SWSs can perform automatic discovery, composition and execution across heterogeneous users and domains. (Dutta 2008, 48.)

SWSs are widely discussed technology in the field of SW. The growing need of web based services has brought the concern of integrating Web Services into SW. SWSs are key technologies of Semantic Web that enable computation over the web (Stollberg & Feir & Roman & Fensel 2011, 3). The integration of Web Services with SW requires the use of languages such as RDF and OWL (Gibbins & Harris & Shadbolt 2011, 2). The SWSs can be used as automated Web Services. Some of the main working areas of automated SWSs are discovery and selection, composition, conversation validation, mediation, and execution support. By utilizing the vision of SW, the main goal of SWSs is to turn internet from an information repository for a human consumption into a World Wide system distributed Web computing. (Stollberg et al. 2011.)

SWSs describe the capabilities and contents on the web in an unambiguous and computer inter-operable language. In addition, it enhances the quality and robustness of tasks such as Web service discovery and invocation. With the application of SWSs, a broad range of automation tasks such as automated composition, interoperation, execution monitoring, and recovery can be performed. (Mcllraith & Martin 2011, 90-91.) Exploring the importance of SWSs particularly in the context of business and services, Dutta (2008, 48-49) argues that SWSs are the extension of Web Services with an explicit representation of meanings. In addition, due to its capacity of automatic discovery, composition, and execution of Web Services, SWSs can change the way knowledge and businesses are provided on the Web today.

There are many prominent technologies which are used for developing the frameworks for SWSs. Some of the major technologies are OWL Service Ontology (OWL-S), Web Service Modelling Ontology (WSMO), and First-order Logic for SWS (FLOWS). However, OWL-S and WSMO are well researched technologies for developing frameworks for SWSs. (Dutta 2008, 50.)

OWL-S is an OWL-based Web service ontology which provides a core set of mark up language to the Web service providers. This mark up language is used for describing the properties and capabilities of Web services. (Dutta 2008.) OWL-S has rich semantics and logical constraints between input and output parameter of services. These help software agents select and operate with the Web Service that meets the requirements of performing tasks. (Vergara & Villagra Berrocal 2011, 5.) WSMO describes various aspects of SWSs and provides the ontological specification for the core elements of SWSs. Some of the major design principles of WSMO are web compliance, ontology-based, goal-driven, strict decoupling, centrality of mediation, ontological role separation, description vs. implementation, and execution semantics. OWL-S works as an ontology and a language that describe Web services while WSMO serves basically as a conceptual model for the core elements of SWSs. (Dutta 2008, 53.)

SWSs seem to be a prominent technology in the field of Semantic Web. Moreover, since the need of Web services is growing rapidly today, the applications of SWSs are very spontaneous. Integration of SWSs in the SW can enhance the overall reasoning capacity of SW.

5 SEMANTIC WEB TOOLS

There are several Semantic Web tools available which can be used to develop Semantic Web applications. It seems that many of such available tools are research prototypes which do not keep up to the standards of commercial solutions. One of the popular website that contains comprehensive lists of such Semantic Web tools is http://semanticweb.org/wiki/Tools/. Similarly the other popular site is http://www.w3.org/wiki/SemanticWebTools .These above mentioned websites contain software of different categories. These different software tools help in different task such as creating and editing ontology, storing RDF, and reasoning engines. Some of these tools have been mentioned below with a precise description.

5.1 Protégé

Protégé is an ontology editor developed by the Stanford Centre of Biomedical Informatics research in collaboration with The University of Manchester. This software is one of the most popular open source software which is widely used for editing ontology. The software is based on java and easy to use. Protégé supports RDF and also has OWL editor within it. There are two built-in reasoners in this software which are FaCT++ and Pellet. Ontology can be modelled in two different ways by using Protégé. These two methods are Protégé Frames methods and Protégé-OWL methods. The former one gives users a chance to build and populate frame-based ontologies. These frame-based ontologies are built following the Open Knowledge Base Connectivity protocol (OKBS). The frame-based ontologies consist of a set of classes, slots and instances. The classes are organized in a general hierarchy to represent the concepts of the proposed domain. The slots are always associated with the classes and they describe the properties and relationships among the classes. The instances hold the specific values associated with class properties. The Protégé-OWL editor is used for creating ontologies for semantic Web. Such ontologies may include the description, properties, and instances of the classes. Figure 12 shows the user interface of the Protégé with a few classes created in it. (Stanford Center for Biomedical Informatics Research 2011.)

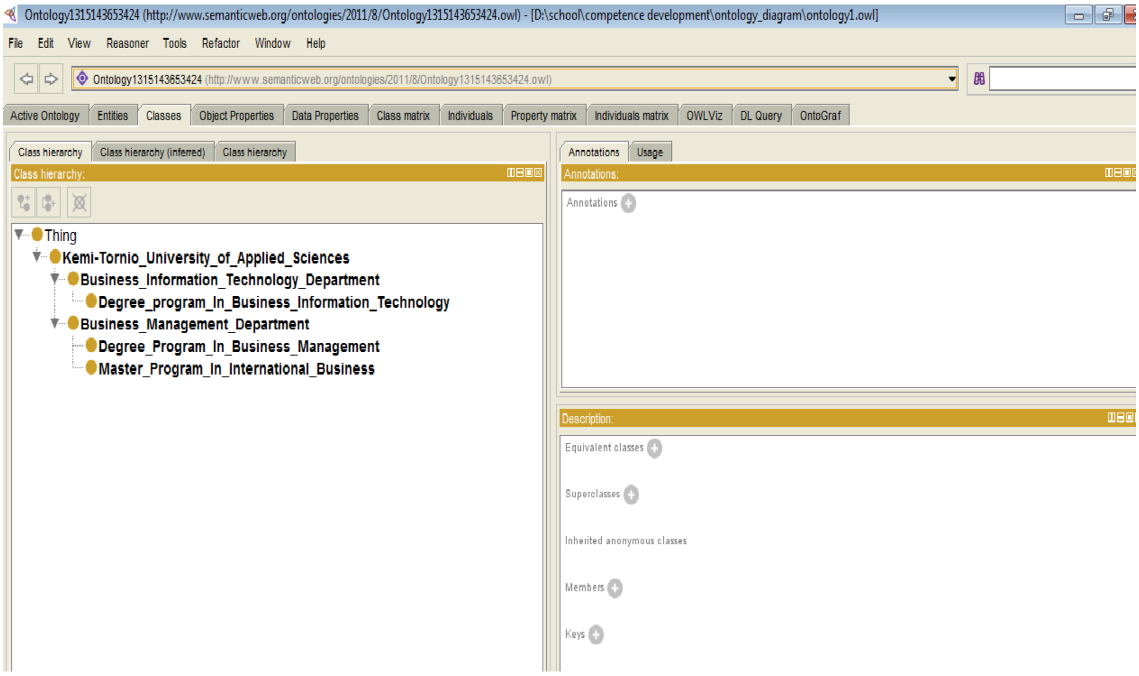


Figure 12. The user interface of Protégé 4.1

5.2 Jena

Jena is a very popular framework tool developed under The Apache Software Foundation (ASF). Jena is widely used to build different Semantic Web applications. This framework is based on Java programming language and has rich collections of tools and Java libraries. The built-in tools and libraries of Jena are mostly used to develop Semantic Web and linked-data applications. Therefore, Jena can be very effective for building Semantic Web Portal. Jena consists of an API that can read, process, and write RDF data in XML, N-triples and turtle formats. Moreover, it has an API and inference engine that handle OWL and RDF ontologies and reasons with the associated data sources. This framework tool includes a strong query engine which is compliant with the latest SPARQL specification. Apart from that, the framework also has a server that can be used to publish the RDF data using a variety of protocols, including SPARQL. (The Apache Software foundation 2012.)

5.3 OntoStudio

OntoStudio is a commercial modelling environment developed by a German company named Ontoprise. This software is used for the creation and maintenance of Ontologies. The OntoStudio seems to be widely used commercially for creating and maintaining ontologies. This tool also provides graphical mapping tool, additional plug- ins, the capacity of editing OWL, RDF/S, RIF, SPARQL, and Object Logic ontologies. Such powerful built-in tools and functions have made the OntoStudio popular among several ontology developers. (Ontoprise 2012.)

5.4 Racer, FaCT++, and Pellet

Racer is a very useful tool for Semantic Web. Racer functions as core inference engine in Semantic Web. It can work with various kinds of ontology editors such as Protégé and other visualization tools (Haarslev & Möller 2012.) FaCT++ is also a very popular open source reasoner developed by the University of Manchester. FaCT++ is written in C++. This software product is available under GNU public license. OWL DL and OWL 2 DL are supported by this FaCT++. The FaCT++ was developed by improving the well established FaCT OWL-DL reasoner. Therefore, the FaCT++ still uses the algorithms used in FaCT OWL-DL reasoner (FaCT PlusPlus 2011.) Pellet is a Java based OWL reasoner. Though this product is open source product, it is also commercially supported by Clark & Parsia LLC. The Pellet seems to be a good choice for the systems that require strong OWL DL reasoning. The Pellet supports OWL 2 profiles which also includes OWL 2. (Clark & Parsia 2011.)

5.5 Sesame

Sesame is an open source RDF framework which uses Java programming language as its main coding language. This framework supports inference and query. RDF schema is used for inference and SPARQL for query purpose (Open RDF Sesame 2011.) Sesame provides a base architecture for storing and querying a large quantity of metadata in RDF and RDF schema. Due to its independent design and implementation properties, Sesame can work on top of various storage devices. Sesame is considered to be the first publicly available implementation of a query language that wisely uses the RDFS semantics. Sesame is a server based application. Thus, it can also function as a remote service for storing and querying data on the Semantic Web. The information available in the form of RDF and RDF schema can be exported independently by using Sesame. (Broekstra et al. 2012.)

5.6 D2RQ

D2RQ is an RDF based platform that is used to access the content of relational databases without having to replicate it into an RDF store. The D2RQ is open source software published under the Apache license. This platform uses SPARQL for executing query on non-RDF databases. The linked data over the Web can also be accessed via D2RQ. Further, this platform has also built-in Apache Jena API which helps retrieve or access information in non-RDF databases. Some other important aspects of D2RQ are: D2RQ mapping language, D2RQ engine, and D2R server. The D2RQ mapping language is declarative enough to describe the relation between an ontology and relational data model. The D2RQ engine works as a plug-in that uses mappings and rewrites the SQL queries called by Jenna API against the databases. The D2R server works as HTTP server. This server provides linked data view and HTML view. (D2RQ 2012.)

6 SEMANTIC WEB PORTALS

Today, Web Portals are widely used as a platform for presenting and exchanging information over the internet. Web Portals have become a very common source for communication and information sharing. Due to the current Web functionalities, the communication and information exchange over the Web has been strengthened not only within the community but also with external communities or individual users. There seems to have appeared different Web Portals with the purpose of providing an open and effective communication forum for their members. The core functionality of a Portal is to collect and present relevant information for the community. The users of such Portals can also publish events or information to the community. Users can locate interesting information in the Portal according to their personal preferences, topic, etc. In addition, users can build their own specific community inside the general community to submit and share information about a certain topic. (Lausen et al. 2004.)

Some of the serious limitations of the present Web technology (pointed out by Lausen et al.2004) are searches are imprecise, often yielding matches to many thousands of hits, and users face the task of reading the retrieved documents to extract the desired information. Therefore, it has been a difficult and time consuming task to search, access, extract, interpret and process information in existing Web Portals. Today, Semantic Web technologies are used to overcome such limitations. Generally, the Web Portals constructed by using Semantic Web technologies are known as Semantic Web Portals. Though the Semantic Web is yet to be applied universally, there are many existing examples of Semantic Web Portals. HealthFinland and MuseumFinland are two major projects of Semantic Web Portals being conducted in Finland. These both Semantic Web Portals are briefly discussed below to illustrate the concept, application, and impact of Semantic Web Portals in general.

6.1 HealthFinland

HealthFinland is a Semantic Web Portal available at http://www.thl.fi/en_US/web/en. With the help of Semantic Web technologies, this Portal offers cost-effective and distributed content creation in an interoperable way. The Portal aggregates the contents automatically and provides the end-users with intelligent services. Moreover, the Semantic Web technologies of this Portal include ontology services and tools. These technologies together enable the collaborative publication of contents which eventually reduce the duplicate work. Figure 13 shows a user interface of HealthFinland.

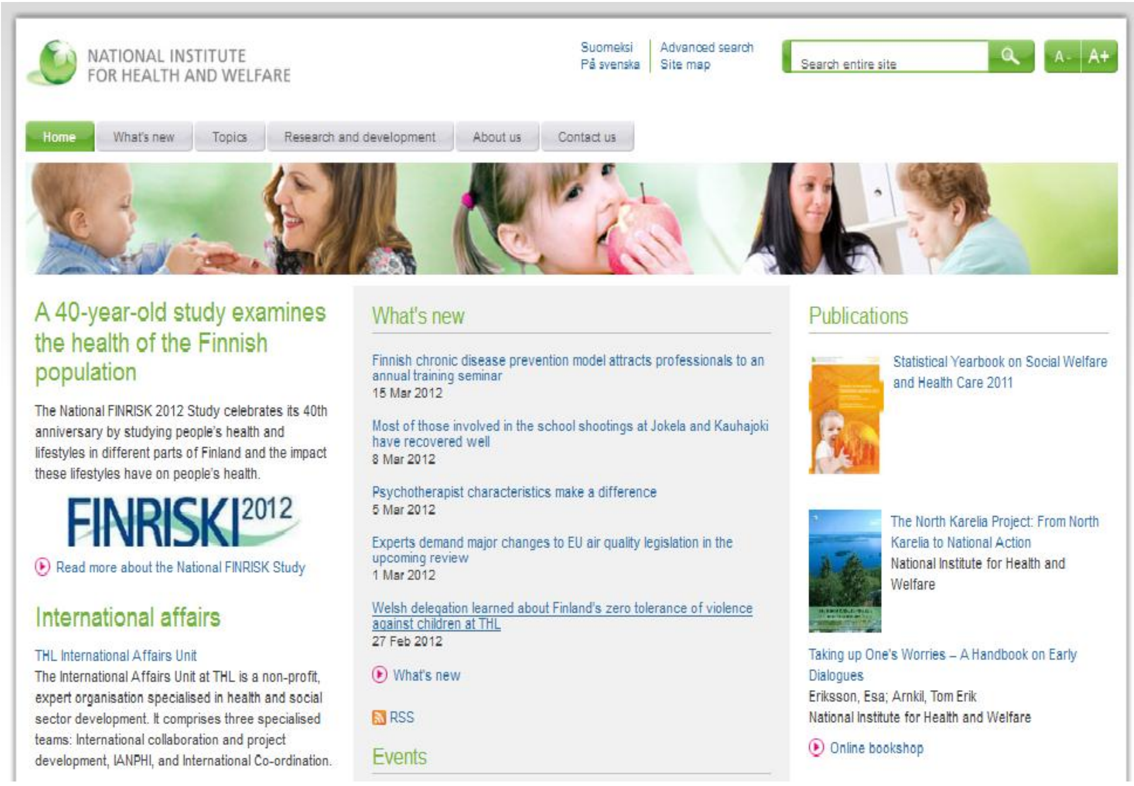


Figure 13. The user interface of HealthFinland (THL 2012)

To perform the internal functions, Semantic Web Portal in general uses a set of vocabularies referenced by metadata. Metadata is description of data. In addition, the ontologies describe all the relevant concepts of the Portal. Thus, such vocabularies and ontologies together describe the document subject matter, content genres, and target audiences. The HealthFinland also uses the same approach. The ontologies in HealthFinland describe a large number of possible topics such as diseases, treatments and anatomy. It also describes the living habits such as diet, exercise, and substance use. The HealthFinland has three core subject domain ontologies that describe its Web contents. These ontologies are: (1) The Finnish General Upper Ontology (YSO), (2) The International Medical Subject Headings (MeSH), and (3) The European Multilingual Thesaurus on Health Promotion (HPMULTI). (Suominen & Hyvönen & Viljanen & Hukka 2009.)

The YSO is achieved by transforming the General Finnish Thesaurus into RDF/OWL format. During the development of HealthFinland, ontology developing/editing software named Protégé was used to achieve the task of the transformation. The YSO contained approximately 20,000 of concepts. The MeSH ontology contained approximately 23,000 of concepts and was transformed into Simple Knowledge Organization System (SKOS) core format without changing the semantics of the vocabulary. SKOS is a standard way of representing knowledge organization system with the support of RDF. The HPMULTI contained approximately 1200 of concepts mostly about health promotion. The HPMULTI was also transformed into SKOS/RDF as MeSH. The subject matter of the HealthFinland was covered using these three vocabularies. (Suominen et al. 2009.)

The HealthFinland emphasizes on three main ideas which are: (1) minimizing duplicate redundant work and costs in creating health content by collaboration, (2) minimizing the content maintenance costs of Portals by allowing the computer to deal with the semantic link maintenance and aggregation of content from various publishers, and (3) providing the end users with intelligent services. These services include searching the right information based on users' own conceptual view to health, and browsing the contents based on their semantic relations. There are three major components in the HealthFinland that fulfil the above mentioned ideas. These three components are Domain Vocabularies, Content Production System, and Semantic Portal. Different aspects of health related information such as topics, genres and audiences, are described by the Domain Vocabularies. The Content Production System has some built-in tools and specifications that can annotate, harvest, and verify the contents. The Semantic Portal, by the help of semantic search and browsing services presents the contents for human users. (Suominen et al. 2009.) These three main components are interlinked. Figure 14 shows how these components are linked with one another.

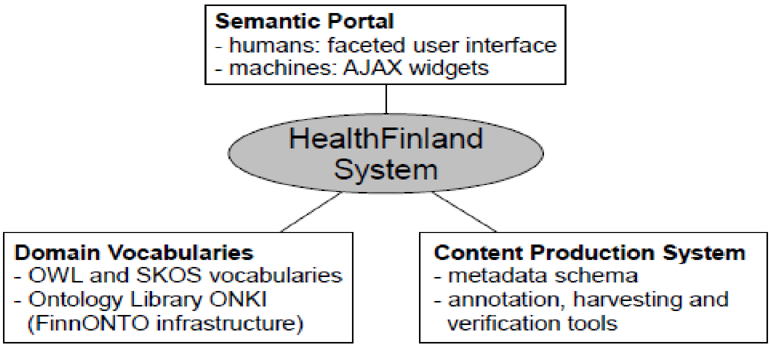


Figure 14. The main components of HealthFinland (Suominen et al. 2009, 4)

Figure 14 also illustrates other sub components of the three main components. The Semantic Portal contains faceted user interface and Ajax widgets by the help of which the data is published on the Portal in human readable form. The FinnONTO infrastructure under the Domain Vocabularies helps interlink the vocabularies and ontologies. The Content Production System includes metadata schema, and tools for annotation, harvesting and verification of data.

6.2 MuseumFinland

MuseumFinland is a Semantic Web Portal available at http://www.museosuomi.fi/ that publishes heterogeneous amount of museum collections. Since this Portal was built as a project of University of Helsinki, further information about this Portal can also be found at http://www.seco.tkk.fi/applications/museumfinland/. The ontologies used in MuseumFinland make the vast amount of collections semantically interoperable. As a result, the museum visitors can experience intelligent content based search and browsing services to the global collection base. Figure 15 shows the user interface of MuseumFinland.

Figure 15. The user interface of MuseumFinland (Semantic Computing Research group 2012)

MuseumFinland contains a wide range of collections. Therefore, this Portal uses various domains' ontologies to cover such large collections. MuseumFinland is based on seven major domain ontologies. These ontologies are Artefacts ontology, Materials ontology, Actors ontology, Situations ontology, Locations ontology, Times ontology, and Collections ontology. All these ontologies contain relevant information to their topics. As for example, the Artefacts ontology contains collections of tangible objects such as pottery, clothes, weapons etcetera. Similarly, the Materials ontology contains artefact materials such as steel, silk, and trees. The Actors ontology contains information about person, company etcetera. The Situations ontology covers tangible happenings, situations, events, and other processes such as farming, feasts, sports, and war that happen in society. The Locations ontology represents information about continent, country, city, and farm. However, in the ontology used in MuseumFinland, locations refer to local places such as Helsinki or Finland. The Times ontology represents different predefined historical periods such as Middle Ages and World War II. The Collections ontology classifies the overall collections of the Portal indicating the name and hosting museum of the collections. The creation and edition of such ontologies were performed by using the ontology editor, Protégé. (Hyvönen et al 2005.)

The content creation process of the MuseumFinland includes the integration of four different databases located in Espoo, Helsinki, and Lahti. Since these databases use different database systems such as Ingress and MS Access, these databases are transformed into a single global database which uses RDF format and remains interoperable syntactically and semantically. Figure 16 shows the content creation processes in detail.

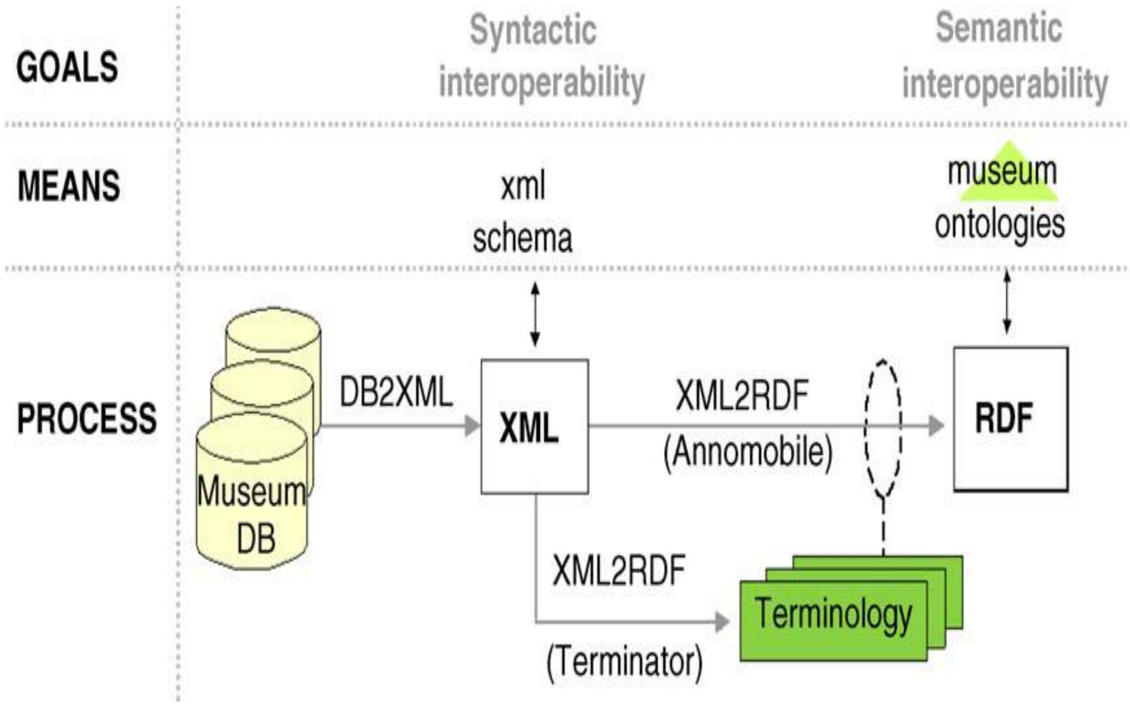


Figure 16. The content creation process in MuseumFinland (Hyvönen et al. 2005, 6)

In Figure 16, the content creation processes involve three major parts. First, the relational database records are transformed into a shared XML language. The arrow DB2XML depicts this transformation process. This transformation gives XML cards as output. The second part defines terminology in RDF. This definition is called term cards. These term cards are used to map XML literals onto URIs in the museum ontology. This transformation is achieved by using a tool named Terminator. The lower arrow with a name XML2RDF depicts this process. The third part of the content creation process is to obtain the semantic interoperability which is achieved by transforming the XML cards into RDF form. The term cards help do this transformation. The upper right arrow with a name XML2RDF represents the process. Thus, finally a set of RDF cards is produced as output which is used in the museum ontology. (Hyvönen et al 2005.)

With the wise use of ontology and logic, the MuseumFinland stands as an example of a SWP that demonstrates how the interoperability of heterogeneous information is possible due to Semantic Web technologies. The use of ontology helps define the concepts precisely and in machine understandable way. Additionally, the Terminological interoperability of the MuseumFinland maps different terms used in different situations into one common shared ontology. This mapping helps tolerate terminological variance by providing the local ontology term conventions to the global ontology. Ontology sharing facilitates with the exact references to the concepts and terms of the external world. Since the MuseumFinland seems to be a practically successful application of Semantic Web Portal, this Portal received the second prize of the Semantic Web Challenge Award 2004. (Hyvönen et al 2005.)

7 ONTOLOGY FOR FINNISH HIGHER EDUCATION SWP

Ontology is the backbone for creating the overall information architecture of Portal SWP. A large number of concepts can be combined and linked with one another using ontology. Ontology development is the first and very crucial step to be taken while developing the SWP. Several concepts are defined and linked with one another which eventually determine the desired knowledge domain. These linked concepts, with the help of the relationships established among them, form the fundamental information architecture for the whole SWP.

This thesis work aims to propose a basic ontology for the EducationFinland SWP - the Semantic Web Portal for Finnish higher education. This ontology is based only on the information from the Kemi-Tornio University of Applied Sciences (KTUAS) due to the time limitations of thesis research. Nevertheless, the objective of this ontology design is to show the way such ontology can be created and applied. The information architecture may vary in other Universities of Applied Sciences in Finland as the degree programs, number of faculty members, and other infrastructure may be different. During the development of this ontology, the ontology editor Protégé was used. Some of the major tasks carried out during the development process were defining concepts/entities, classes, and their hierarchy, defining entity relationships, defining data and object properties, specifying domain and range of the entities. The hierarchy of the classes was built first. Figure 16 includes a screenshot with the hierarchy of classes built for EducationFinland.

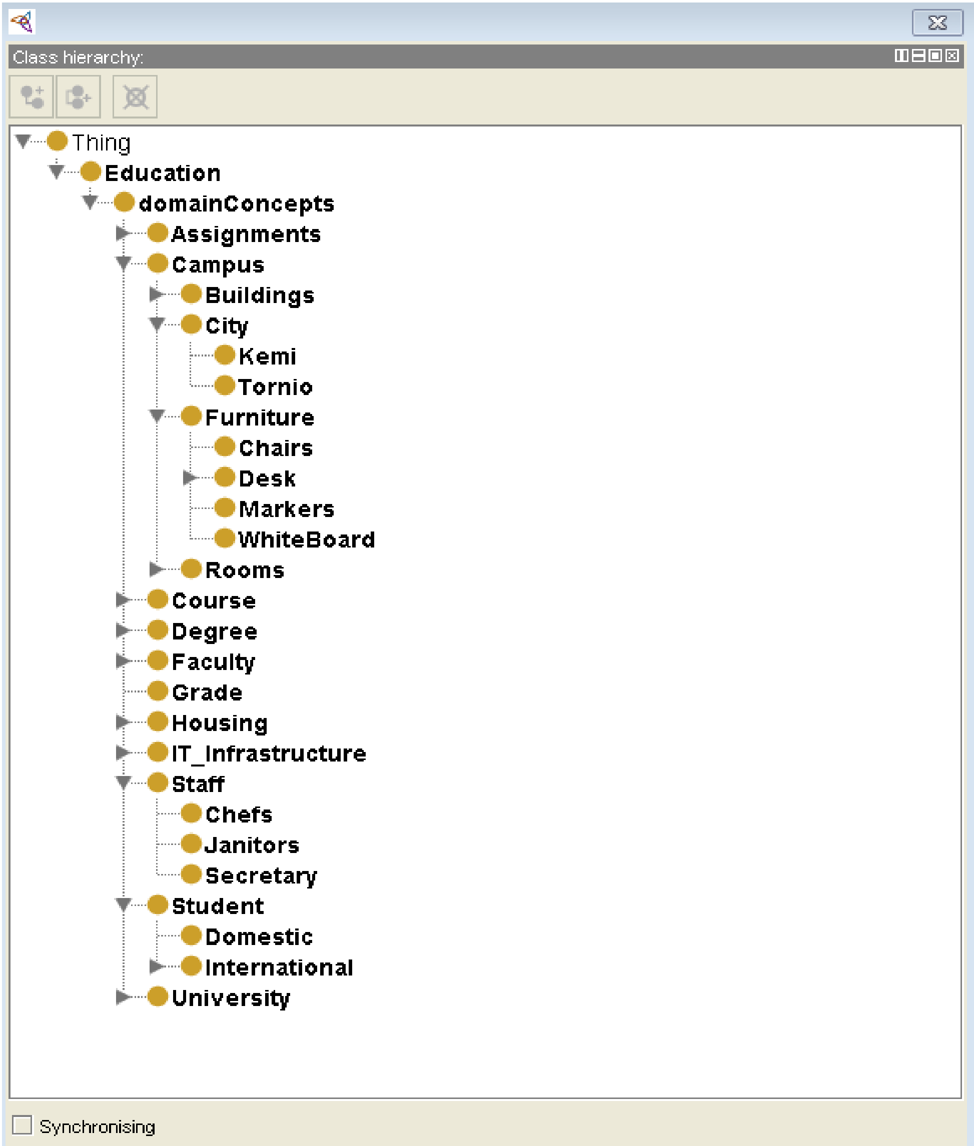


Figure 17. The class hierarchy of the KTUAS Ontology

A class named "Thing" is on the top of the hierarchy. This is a universal class for any ontology that is created by Protégé. The Protégé uses this class as the universal class and always places it on top of the hierarchy. In practice, when the user wants to create a hierarchy, the class "Thing" is automatically created and is always present. All other classes created will be subclasses of the class "Thing". The vertical line in Figure 16 represents the hierarchy line. All the classes that stand on the same line represent the same hierarchy level. While going to lower hierarchy, a space is created and another hierarchy line is drawn to the right. One class can have several objects in it. The arrow represents the class and the objects are listed under the class with a round yellow dot preceding their names. The arrow pointing downward represents the expanded form of a class which means all the objects under such class can be seen visually. The arrow pointing to right is a non-expanded arrow and it visually hides the objects under it.

The objective of creating a class hierarchy is to structure the entities of the domain in a certain format. EducationFinland, several entities such as campus, course, degree, student, faculty, and staff are put under different classes. After creating the class hierarchy, object properties and data properties are defined for each object. The data properties are used to define what kind of data value (integer, string) the object can hold. The object properties are used for defining the links among various objects. Figure 17 shows the sample object properties defined for the EducationFinland's ontology.

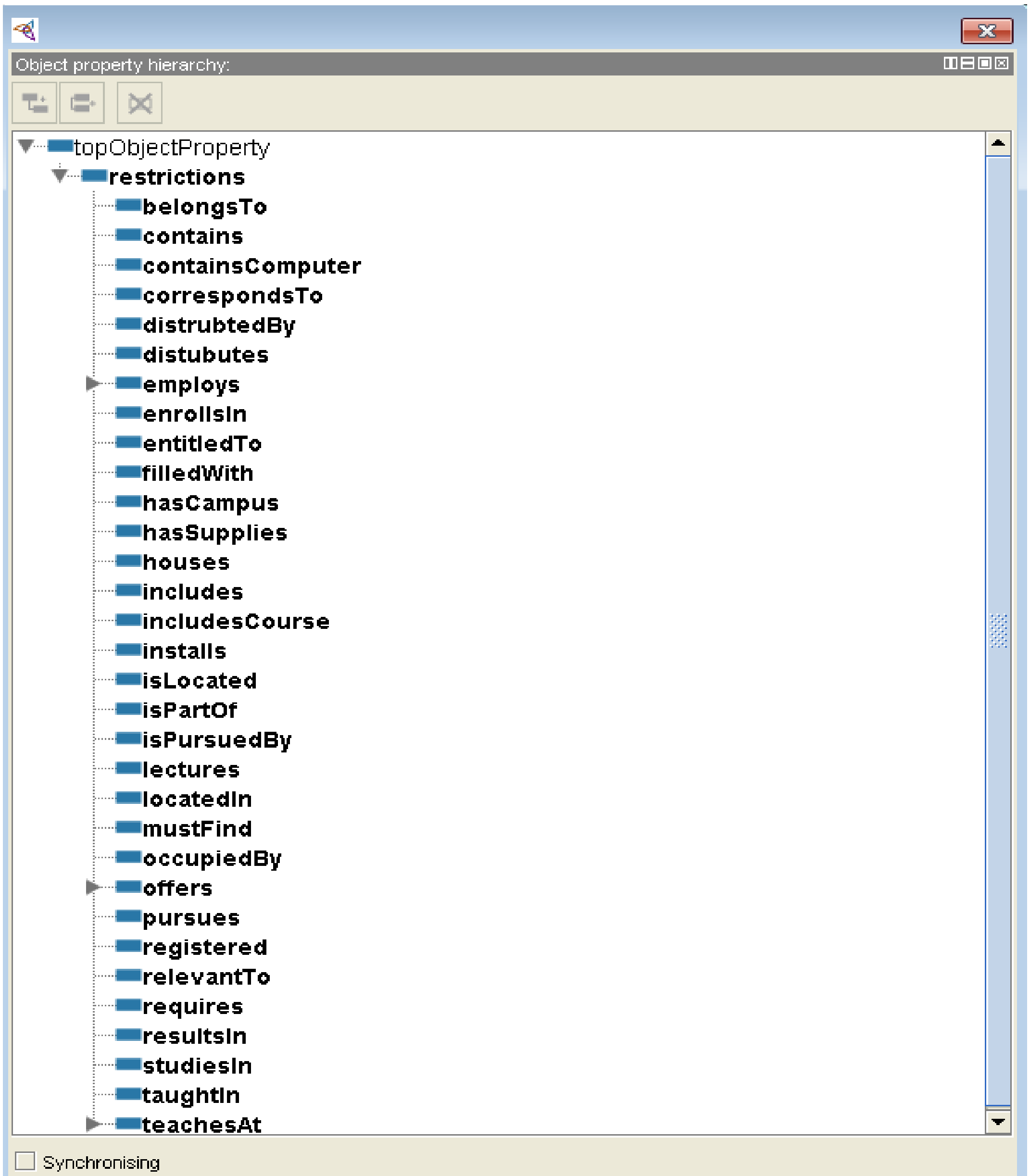


Figure 18. Object properties of the KTUAS Ontology

The object properties are defined to establish relationships among classes. To avoid the ambiguity in describing the relationships among classes, the describing terms used as object properties are chosen carefully. While combining two or more words as an object property, the standard rule of ontology is followed. Based on that rule, the first word is used starting with lower case and the rest word/s starting with upper case. There is no space used between these words. As for example, enrollsIn is used as an object property to establish relation between “student” and “course”. The Protégé offers a visual interface for viewing the ontology. An additional plugin OWL Viz is used for the visual interface. Figure 18 shows the visual image of the EducationFinland’s ontology.

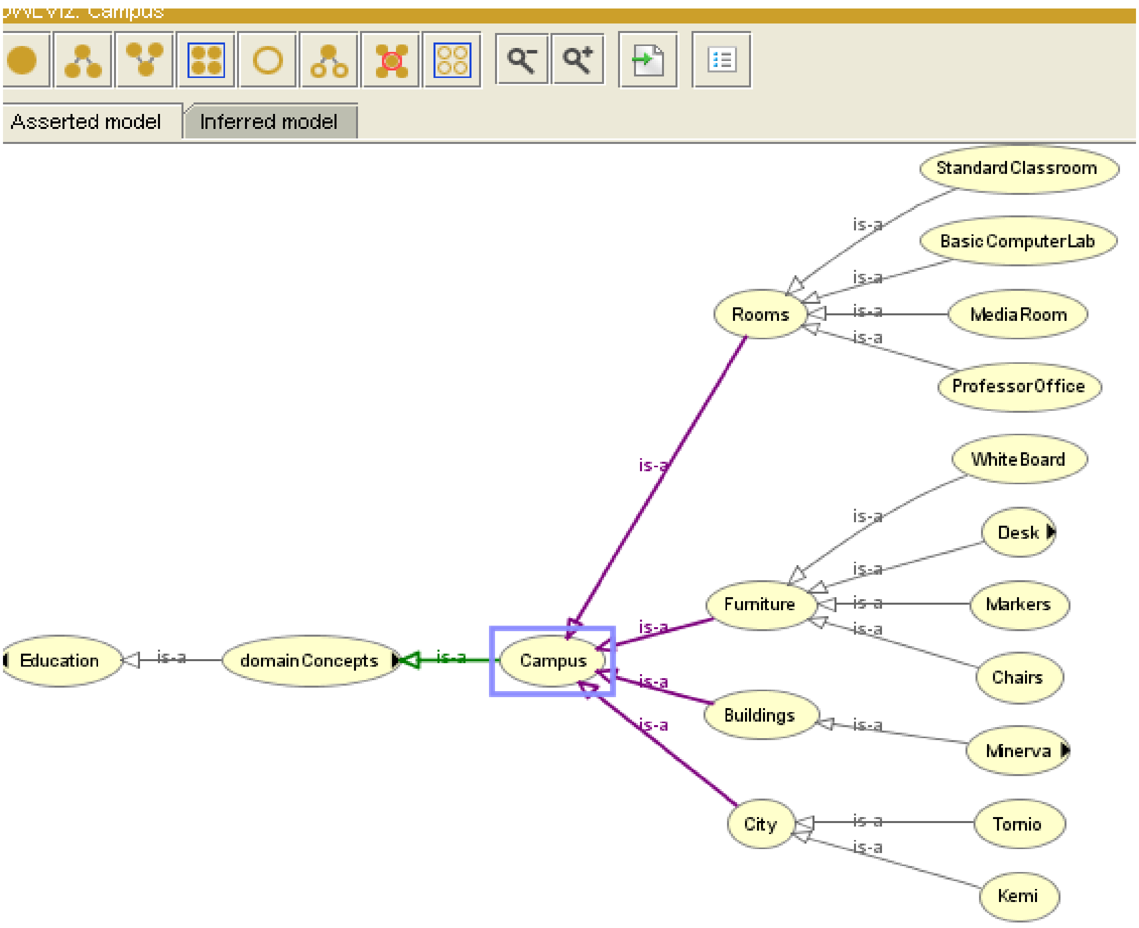


Figure 19. Visual representation of the KTUAS Ontology

The OWL Viz plug-in has compatibility issues with Protégé which is older than version 4.1. One reason to use Protégé 4.1 for creating the ontology is the ability to use OWL Viz. When a class is chosen from the class hierarchy in Protégé, the corresponding class is highlighted with a rectangle in the OWL Viz window pane, and all the other lower classes are presented in their expanded form. However, the classes which are up in hierarchy from the selected class are presented in non-expanded form.

Though the OWL Viz presents an appropriate visual interface for viewing the hierarchical architecture of ontology, it still lacks the ability to somehow present the defined object properties. When visualizing ontology, it is also important to make sure that the classes are linked with correct object properties. To achieve this goal of visually checking if the classes are linked with the correct object properties, another plug-in for Protégé named OntoGraf is used. The OntoGraf presents ontology in graphical form and with the help of the built in reasoner capacity, it also detects and presents the object properties. The object properties become visible when mouse is hovered on the linking arrow of classes. Figure 19 shows the ontology presented in OntoGraph with an example of an object property appearing on a mouse hover.

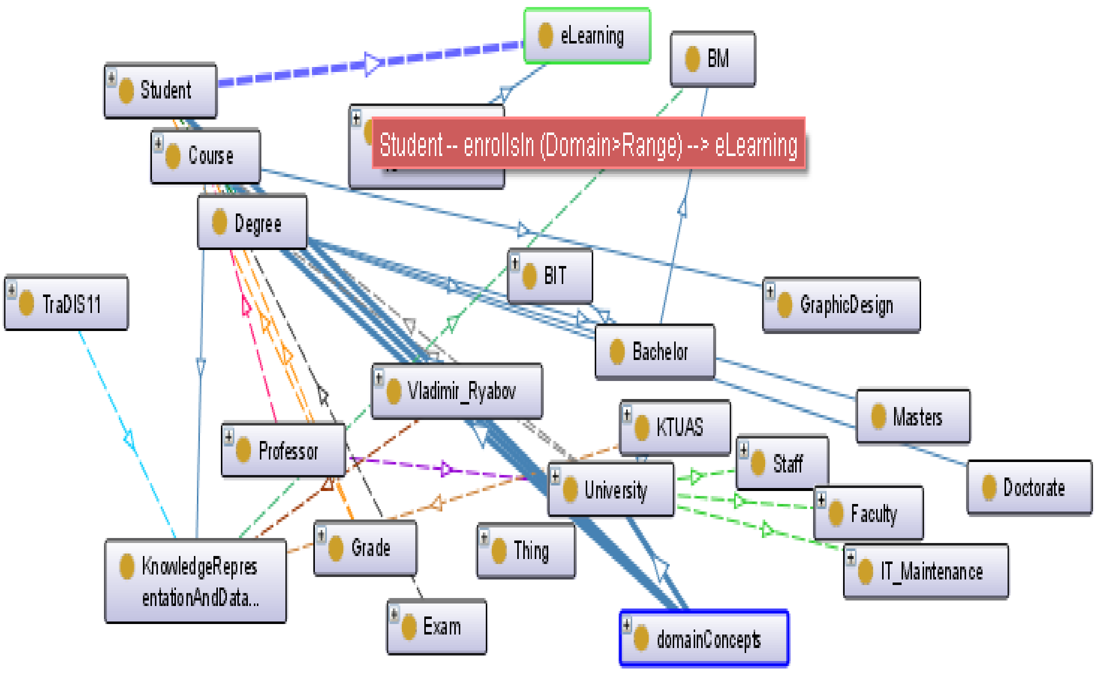


Figure 20. The KTUAS Ontology presented by OntoGraf

The OntoGraf presents the KTUAS Ontology in a graphical form in which different classes are linked with one another. The linking arrows hold the values of object properties. Hovering mouse on the linking arrows pops up the corresponding object properties. As for example, an object property is presented under the red background in

Figure 19. The linking arrow in Figure 19 that represents the popped up object property connects the classes “eLearning” and “Student” with the object property “enrollsIn”.

Creating ontology is a challenging task and requires a significant amount of brainstorming and pieces of advice from experts. The KTUAS Ontology presented in this thesis covers only limited number of concepts with a purpose of serving as an example. Therefore, this ontology can be expanded enormously with a large number of additional concepts. However, the basic rules applied in the development process of this ontology remain the same. One of the goals of creating and presenting this ontology is also to show the applicability and efficiency of Protégé as an ontology developing tool. The class hierarchies and the object properties in this ontology serve as general example on behalf of Finnish higher education. While being specific to any University of Applied Sciences, these class hierarchies and object properties can be adapted accordingly.

The objectives of this thesis work are to study the concepts of Semantic Web Portals, required tools and techniques with explanation of their roles in building Semantic Web Portals, and to create sample ontology for Finnish higher education. Creating a fully functional Semantic Web Portal is a large project that requires plenty of resources and time. It is practically not possible to create fully functional SWP within the time frame allocated for this thesis work. Therefore, a sample ontology is presented as a practical part of this thesis work besides the descriptions of the concept of Semantic Web Portal, and required tools and techniques for building Semantic Web Portal. Protégé is not a compulsory tool to be used for developing ontology. There are several such other tools available commercially together with open source tools. Some of these tools have been discussed in chapter 5. However, the study shows that Protégé is available free of cost and is widely used for developing ontology. Therefore, it was selected for developing the KTUAS Ontology for this thesis work.

Since ontology serves as initial and core architecture of SWP, the KTUAS Ontology can give an in-depth insight for designing a Semantic Web Portal for Finnish higher education. Based on the KTUAS Ontology, further development processes of building Semantic Web Portal can be initiated.

8 EDUCATION FINLAND

This chapter presents a general description of information architecture, development processes, and functionalities of EducationFinland. The roles of the KTUAS Ontology in building fully functional Semantic Web Portal EducationFinland are discussed. Since practical implementation of EducationFinland requires a huge amount of time as well as experts' work, the actual implementation cannot be achieved in this thesis. However, deriving from this thesis work, a team of education professionals from different institutions of Finnish higher education can achieve the actual implementation of EducationFinland. Based on the gathered information for building EducationFinland and the KTUAS Ontology presented in chapter 7, the possible information architecture, user interface, and methods of user interactions can be suggested. HealthFinland and MuseumFinland discussed in chapter 6 can be closely followed while making above mentioned suggestions.

The core design of information architecture of Semantic Web Portal is built through ontology. Developing ontology is the first practical step in the process of developing Semantic Web Portal. Therefore, the further development of EducationFinland can be carried out by utilizing the sample KTUAS Ontology presented in this thesis. The KTUAS Ontology contains only limited number of concepts as it presently serves as an example. However, during the actual implementation, this ontology can be extended by adding a large number of concepts. Such concepts can be collected from various institutions involved under the Finnish higher education system. As these concepts will serve as common concepts for EducationFinland, the professionals from different institutions involved in Finnish higher education should also agree upon them. The ontology developing tool named Protégé was used to build the sample KTUAS Ontology. Therefore, the same tool is suggested to be used for the extension of the KTUAS Ontology during the actual implementation.

As the sample KTUAS Ontology is OWL/XML ontology, it is advisable to use OWL/XML ontology during the actual implementation rather than using RDF/XML ontology. The OWL/ XML ontology, being increasingly expressive and scalable, will be easier to get familiar with and accommodate complex and greater number of concepts. As the approach used by HealthFinland described in section 6.1, the internal functions of EducationFinland can be performed by using a set of vocabularies referenced by metadata. These vocabularies and the KTUAS Ontology can describe the document subject matter, content genres, and target audiences.

Multiple ontologies can also be used if proposed by education professionals of Finnish higher education. In this case, all these ontologies should be transformed into a single format, preferably in RDF/OWL format and should be combined and put into a single new ontology. This new ontology will function as the core ontology for EducationFinland. The transformation of ontologies can be achieved by using Protégé tool. In the development process of EducationFinland, the next step after developing the ontology will be to create three other main components: domain vocabularies, content production system, and semantic Portal. Information related to topics, genres and audiences will be described by the domain vocabularies. The Content Production System will have built-in tools and specifications that will annotate, retrieve, and verify the contents. The semantic Portal will browse and publish contents for users.

The user interface can be made to some extent similar to the interface of MuseumFinland or entirely different than that. However, this decision should be solely based upon the agreement among the education professionals of Finnish higher education. In either case, the user interface should satisfy some basic design and functionalities. The basic design and functionalities include search function, categorized and hierarchical form of information. The user interactivity should be made as simple and natural as possible. To achieve the goal of enhanced user interaction, the search function should be enhanced semantically.

EducationFinland, being a Semantic Web Portal, will work differently in comparison with any ordinary Web Portal. Such differences will mainly exist in the inner information architecture, data storage and data retrieval mechanism, and data publishing. However, from the user interaction point of view, such differences will be depicted from the way the search engine and data publishing work. Google and other search engines publish the list of links by matching the characters of search keywords with metadata of Web pages. However, the use of semantics in EducationFinland will make its search engine function differently. In EducationFinland, the semantics of the search keywords will be understood by the machine and relevant information will be published rather than just the list of links.

Education professionals of Finnish higher education may also decide to use multiple databases for storing data. In this case, all these databases should be integrated into one global database. This global database will support RDF format and will be semantically and syntactically interoperable. The interoperability is important as the changes in local databases should also reflect in the global database. If the local databases use different database systems, a similar database integration process described in section 6.2, depicted from Figure 15, can be used. With the help of this database integration process, even local databases that use different database systems can be integrated into the global database providing a common RDF data format without changing the semantics of the data.

D2RQ discussed in section 5.6 can also be utilized to access the content of relational databases without having to replicate the content into RDF store. The use of D2RQ will also help access linked data over the Web. If the EducationFinland requires to access non-RDF databases, the built-in Jena API of D2RQ can be utilized to access such databases. Since there are many member institutions of Finnish higher education, there is a great possibility of having multiple databases, database systems, and data formats. Therefore, the use of D2RQ would be a good choice for the development of EducationFinland.

The decision of choosing tools and techniques of building EducationFinland will entirely rely on the agreement among the education professionals who will be involved in the development project of EducationFinland. Thus, the tools and techniques suggested above may vary slightly or greatly depending on what these project team members decide. Nevertheless, this thesis work provides solid background information about the concept of Semantic Web and Semantic Web Portal together with the description about numerous widely used tools and techniques. As a result, there are plenty of opportunities to choose from different tools and techniques described in this thesis work. To conclude, utilizing this thesis work as foundation if further research is carried out through a team of education professionals of Finnish higher education, there is strong possibility of actual implementation of EducationFinland.

9 CONCLUSIONS

The main objectives of this thesis work are to explore the tools and technologies used for building and implementing SWPs, present a sample ontology representing Finnish higher education system, and briefly describe the idea of building a SWP for Finnish education system.

This thesis work, keeping the Semantic Web Portal for Finnish education as a core focus, studied the concept and information architecture of Semantic Web Portals. To understand and illustrate the implementation possibilities of Semantic Web Portals, various existing tools and techniques used to build such Portals were studied. As a practical part, the thesis also presented a sample ontology, i.e. KTUAS Ontology, representing Semantic Web Portal for Finnish education. Protégé, an ontology editing tool, was chosen to build the sample ontology. The roles of semantics and its impact on the Web technology were discussed. In addition, how the Semantic Web Portal for Finnish education would function and how it would be increasingly efficient than ordinary Web Portals were discussed.

Designing and implementing a fully functional Semantic Web Portal is a task full of challenges. The concept of Semantic Web with its few functional Portals has been under one of the hot topics for research over nearly a decade. However, the universal adaptation and implementation of Semantic Web Portals seem very limited. The ontology based information architecture of Semantic Web Portal is a recent approach and requires a significant amount of experts' work. Due to the need of greater amount of time and experts' work, designing and implementing a fully functional Semantic Web Portal for Finnish higher education within the time frame allocated for this thesis work is practically impossible. In addition, the successful implementation of Semantic Web Portal for Finnish higher education needs Semantic Web technology experts as well as education professionals of Finnish education to agree upon the design. Therefore, even the proposed KTUAS Ontology in this thesis may need numerous adjustments during its real implementation. Since several institutions are involved in Finnish higher education, the information for the Semantic Web Portal for Finnish higher education should be collected from various institutions. To promote Semantic Web globally, major website companies should switch to the Semantic Web.

Despite the fact that there are some limitations in designing and implementing Semantic Web Portal for Finnish higher education, most of these limitations are possible to be overcome in near future as the research and technological advancements in this field are growing rapidly. Moreover, the use of semantics in Web Portals, with its capabilities of understanding, analyzing, and reasoning data, works as a potential factor for additional research works to be conducted. As a result, the number of semantic Web Portals such as HealthFinland and MuseumFinland is growing constantly. The growing number of commercial and open source tools and techniques for building Semantic Web Portals is another motivational factor that points out the importance of further research and development of Semantic Web Portals. Some of those tools and techniques were explained in chapters 3, 4, and 5.

This thesis contains abundant information about Semantic Web in general, Semantic Web Portal in particular, and a functional ontology representing Finnish higher education. In other words, this thesis not only provides sufficient information about building Semantic Web Portals but also presents sample ontology, i.e. the KTUAS Ontology which is the first step of creating a fully functional Semantic Web Portal for Finnish higher education system. Therefore, based on the information gathered in this thesis and by utilizing the presented sample KTUAS Ontology, further research can be carried out to build a fully functional EducationFinland.

REFERENCES

Printed

Alesso, H. Peter & Smith, Craig F. 2004. Developing Semantic Web Services. A K Peters/CRC Press Limited, 888 Worcester Street, Suite 230 Wellesey, MA 02482.

Antonio, Grigoris & Harmelen, Frank van 2004. A Semantic Web Primer. The MIT Press Cambridge, Massachusetts, London, England.

Berners-Lee, Tim & Hendler, James & Lassila, Ora 2001. The Semantic Web. Scientific American Magazine.

Brietman, K.K & Casanova, M.A. & Truszkowski, W. 2007. Semantic Web: Concepts, Technologies, and Applications. Nasa Monographs in Systems and Software Engineering. Springer-Verlag London Limited.

Colomb, R. M. 2007. Ontology and the Semantic Web. IOS Press, Amsterdam.

Dutta, Biswanath 2008. Semantic Web Services: A Study of Existing Technologies, Tools and Projects. DESIDOC Journal of Library & Information Technology 28(3), 47-55.

Hebeler, John & Fisher, Matthew & Blace, Ryan & Perez-Lopez, Andrew 2009. Semantic Web Programming. Wiley Publishing, Inc. Indiana.

Hitzler, Pascal & Krötzsch, Markus & Rudolph, Sebastian 2010. Foundations of Semantic Web Technologies. CRC Press, Taylor & Francis Group, Boca Raton, London.

Hyvönen, Eero & Mäkelä, Eetu & Salminen, Mirva & Valo, Arttu & Vinjanen, Kim & Saarela, Samppa & Junnila, Miikka & Kettula, Suvi 2005. MuseumFinland—Finnish museums on the semantic Web. Helsinki Institute for Information Technology (HIIT), University of Helsinki, and Helsinki University of Technology, Finland.

Lei, Yuangui & Lopez, Vanessa & Motta, Enrico 2012. An Infrastructure for Building Semantic Web Portals. Knowledge Media Institute (KMi), The Open University, Milton Keynes.

Liang, Yu 2007. Introduction to Semantic Web and Semantic Web Services. Chapman & Hall/CRC, Taylor & Francis Group New York.

Maedche, Alexander & Staab, Steffen 2011. Ontology Learning for the Semantic Web. Institute AIFB, D-76128 Karlsruhe, Germany.

Mahalingam, Kuhanandha & Huhns, Michael 1998. Representing and Using Ontologies. Center for Information Technology, Department of Electrical and Computer Engineering, University of South Carolina, Columbia, SC. U.S.A.

Mäkeläinen, Sami 2005. Social Semantic Web. Paper for ”Tiedonhallinta Semanttisessa Webissä”-seminar, University of Helsinki.

Miltiadis et al. 2005. The Semantic Web. Emerald Group Publishing Ltd.

Parsia, Bijan & Patel-Schneider, Peter 2004. Meaning and the Semantic Web. University of Maryland College Park, MD, USA and Murray Hill, NJ, USA.

Panneerselvam, R. 2004. Research Methodology. Prentice-Hall of India Private Limited, New Delhi.

Segaran, Toby & Evan, Colin & Tayler, Jamie 2009. Programming the Semantic Web. O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

Sheth, Amit & Ramakrishnan, Cartic & Thomas, Christopher 2005. Semantics for the Semantic Web: The Implicit, the Formal and the Powerful. Int'l Journal on Semantic Web & Information Systems 1(1).

Sowa, John 1999. Knowledge Representation: Logical, Philosophical, and Computational Foundations. Brooks Cole Publishing Co., Pacific Grove, CA.

Stollberg, Michael & Feier, Christina & Roman, Dumitru & Fensel, Dieter 2011. Semantic Web Services- Concepts and Tenchnologies. Digital Enterprise Research Institute, University of Innsbruck, Austria.

Suominen, Osmo & Hyvönen, Eero & Viljanen, Kim & Hukka, Eija 2009. HealthFinland, a National Semantic Publishing Network and Portal for Health Information. aSemantic Computing Research Group (SeCo), Helsinki University of Technology (TKK), Department of Media Technology University of Helsinki, Department of Computer Science.

Uschold, Michael 2011. Where are the semantics in the semantic Web. The Boeing Company, USA.

Weller, katrin 2010. Knowledge and information: Knowledge representation in the Social Semantic Web. Walter de Gruyter GmbH & Co. KG, Berlin/New York

Not-printed

Aquin, Mathieu & Schlicht, Anne & Stuckenschmidt, Heiner & Sabou, Marta 2011. Criteria and Evaluation for Ontology Modularization Techniques <http://ki.informatik.uni-annheim.de/fileadmin/publication/dAquin10CriteriaChapter.pdf>

Broekstra, Jeen & Kampman, Arjohn & Harmelen, Frank 2012. Faculty of Sciences, Vrije Universiteit, Amsterdam, The Netherlands. Downloaded 10th March 2012. <http://www.openrdf.org/doc/papers/Sesame-ISWC2002.pdf>

Bruijn, Jos de & Fensel, Dieter & Polleres, Axel 2004. WSML Working Draft.Downloaded 6th December 2011. <http://www.wsmo.org/2004/d20/v0.1/20040621/d20v0.1_20040621.pdf>

Clark and Parsia 2011. Pellet OWL 2 reasoner for Java. Downloaded February 12 2012. <http://clarkparsia.com/pellet>

D2RQ 2012. Accessing Relational Databases as Virtual RDF Graphs. <http://d2rq.org/>

FaCT PlusPlus 2011. FaCT++. Downloaded February 12 2012. <http://owl.man.ac.uk/factplusplus/>

Garcia, Roberto & Gil, Rosa 2012. Improving Human–Semantic Web Interaction: The Rhizomer Experience. GRIHO Human-Computer Interaction Research Group at the Universitat de Lleida, Lleida, E-25001 Spain.

Gibbins, Nicholas & Harris, Stephen & Shadbolt, Nigel 2011. Agent-based Semantic Web Services. Department of Electronics and Computer Science University of Southampton Southampton, United Kingdom. <http://eprints.aktors.org/167/01/p455-gibbins.pdf>

Haarslev, Volker & Möller, Ralf 2012. A Core Inference Engine for the Semantic Web. Concordia University, Montreal, Canada and University of Applied Sciences, Wedel, Germany. Downloaded 11th March 2012. <http://ftp.franz.com/agraph/racer/Racer_whitepaper.pdf>

Information Village 2009. Exploratory Research. Downloaded February 15, 2011. <http://www.info-village.info/exploratory-research/>

Kobie, Nicole 2010. Conrad Wolfram on communicating with apps in Web 3.0 <http://www.itpro.co.uk/621535/q-a-conrad-wolfram-on-communicating-with-apps-in- web-3-0>

Lausen, Holger & Stollberg, Michael & Hernandez, Ruben & Ding, Ying & Han, Sung-Kook & Fensel, Dieter 2011. Semantic Web Portals-State of the Art Survey. Next Web Generation Group, IFI – Institute for Computer Science University of Innsbruck, Austria. Downloaded February 12, 2012.
<http://sw-portal.deri.at/papers/publications/SemanticWebPortalSurvey.pdf>

Mcllraith, Sheila A. & Martin, David L. 2011. Bringing Semantics to Web Services. Stanford University.

Moriya, Netzer 2007. Ontology review. Downloaded 13 October 2011.
<http://www.iqlue.com/Ontology.jsp>

Mostafa, Aref & Zhengbo, Zhou 2011. Ontology Web Language (OWL) for a multi-agent understanding system. Department of Computer Science and Engineering, University of Bridgeport. Downloaded 28th December 2011.
<http://www1bpt.bridgeport.edu/~zhengboz/1001_aref.pdf>

Nakate, Shashank 2010. What is Web 3.0. Last modified June 21, 2010.
<http://www.buzzle.com/articles/what-is-web30.html>

Ontoprise 2012. Semantic Infrastructure OntpStudio.
<http://www.ontoprise.de/en/products/ontostudio/>

Open RDF Sesame 2011.
<http://semanticweb.org/wiki/Sesame>

Selcuk, Candan & Huan, Liu & Reshma, Suvarna 2011. Resource Description Framework: Metadata and Its Applications. Department of Computer Science & Engineering Arizona State University, Tempe 3 (1). Downloaded May 14, 2011.
<http://www.sigkdd.org/explorations/issue3-1/candan.pdf>

Semantic Computing Research group 2012. MuseumFinland.
<http://www.museosuomi.fi/>

Stanford Center for Biomedical Informatics Research 2011. Protege. Downloaded 4th October 2011.
<http://protege.stanford.edu/>

Stroka, Stephanie 2005. Knowledge Representation Technologies in the Semantic Web. Information Technology and Systems Management, Salzburg University of Applied Sciences.
<http://stexx.files.wordpress.com/2010/07/termpaper.pdf>

Tamma, Valentina 2010. Semantic Web Support for Intelligent Search and Retrieval of Business Knowledge
<http://lpis.csd.auth.gr/mtpx/sw/material/IEEE-IS/IS-25-1.pdf>

Tazzoli, Roberto & Castangna, Paolo & Campanini, Stefano 2012. Towards a Semantic Wiki Wiki Web. Downloaded 12 April 2012. <http://www.tecweb.inf.puc-rio.br/semweb/space/Platypus+Wiki/platypuswiki.pdf>

The Apache Software Foundation2012. Apache Jena <http://incubator.apache.org/jena/>

THL 2012. National Institute for Health and Welfare, Helsinki,Finland. <http://www.thl.fi/en_US/web/en>

Vergara, Jorge E. & Villagra, Victor A. & Berrocal, Julio 2011. Application of OWL-S to define management interfaces based on web services. Department of computer engineering, University of autonomous, Madrid, Spain. <http://jungla.dit.upm.es/~jlopez/publicaciones/mmns05.pdf>

World Wide Web Consortium 2004. Web Ontology Language. Downloaded December 2011. <http://www.w3.org/TR/2004/REC-owl-features-20040210/#s1.3>

World Wide Web Consortium 2009. Resource Description Framework. Downloaded December 2011. <http://www.w3.org/TR/2004/REC-owl-features-20040210/#s1.3>

World Wide Web Consortium recommendation 2004. RDF Primer. <http://www.uazuay.edu.ec/bibliotecas/mbaTI/pdf/RDF%20Primer.pdf>

Zarri, Gian Piero 2002. Semantic Web and Knowledge Representation. Centre National de la Recherche Scientifique (CNRS), Paris, France. Downloaded February 2011. <http://ieeexplore.ieee.org.ez.tokem.fi/stamp/stamp.jsp?tp=&arnumber=1045879&tag=1>