Ontologies, Different Reasoning Strategies, Different Logics, Different Kinds of Knowledge Representation: Working Together

Mela Bosch

Document Management Consultant, Milan, Italy, melabosch@europe.com

Mela Bosch is a member of the Research Project on Integrative Level Classification of ISKO Italian Chapter: http://www.iskoi.org/ilc. She is currently working as an independent Document Management consultant in Milan. She is also among the E-learning staff at the Journalism Department, La Plata National University, Argentina. She is also a Technologies in Social Communication researcher for that very same institution: http://perio.unlp.edu.ar/tecnologias/. Her educational background includes training in Linguistics and Software Engineering in Argentina, E-Learning in Milan and Document Management in Barcelona. She has worked as an ordinary professor at the Library Department, La Plata National University, Argentina. Articles and papers can be found at E-Lis: http://eprints.rclis.org/

Mela Bosch. **Ontologies, Different Reasoning Strategies, Different Logics, Different Kinds of Knowledge Representation: Working Together.** *Knowledge Organization, 33(3)* 153-159. 13 references.

ABSTRACT: The recent experiences in the building, maintenance and reuse of ontologies has shown that the most efficient approach is the collaborative one. However, communication between collaborators such as IT professionals, librarians, web designers and subject matter experts is difficult and time consuming. This is because there are different reasoning strategies, different logics and different kinds of knowledge representation in the applications of Semantic Web. This article intends to be a reference scheme. It uses concise and simple explanations that can be used in common by specialists of different backgrounds working together in an application of the Semantic Web.

1. Introduction

Berners-Lee's dream was that information in the Semantic Web would be expressed with "well-defined meanings, enabling computers and people to work in cooperation." However, the activity of reasoning about the real world and the digital world at the same time is a complex one that involves multiple reasoning strategies, structures and elements from different languages. Thanks to the contribution of many disciplines, the Semantic Web is continuously developing. Therefore, the formalisms of knowledge representation consist of different kinds of logics in different kinds of representation formats. There are many aspects involved, such as description logic, classification systems, object oriented data structure and markup languages.

This article is based on the writer's experience as a document management consultant for companies and organizations in Milan, Italy, and Buenos Aires, Argentina, and as researcher and professor of Technologies in Social Communication at the National University of La Plata, Argentina. Having a common scheme proved very useful when working with professionals from different backgrounds. It allowed us to know in which level of knowledge representation we were at any given moment and to understand each other's points of view. The current situation will be outlined with the aid of graphics and brief explanations.

1.1 Knowledge representation in the digital world

Different disciplines and professionals involved would agree with the following definition:

> A knowledge representation (KR) is most fundamentally a surrogate, a substitute for the

thing itself, used to enable an entity to determine consequences by thinking rather than acting, i.e., by reasoning about the world rather than taking action in it.

It is a set of ontological commitments, i.e., an answer to the question: In what terms should I think about the world? It is a fragmentary theory of intelligent reasoning, expressed in terms of three components: 1) the representation's fundamental conception of intelligent reasoning; 2) the set of inferences the representation sanctions; and 3) the set of inferences it recommends.

It is a medium for pragmatically efficient computation, i.e., the computational environment in which thinking is accomplished. One contribution to this pragmatic efficiency is supplied by the guidance a representation provides for organizing information so as to facilitate making the recommended inferences. It is a medium of human expression, i.e., a language in which we say things about the world (Davis et al. 1993).

Many definitions of the concept "knowledge representation" may be put forth. This has been the subject of many discussions throughout the years. This work uses this definition only. Although it may be considered a very general one, it has been chosen because the writer believes it encloses the main features of the problem that occupies this work: the building, maintenance and reuse of ontologies.

2. Knowledge representation in the Semantic Web

When applied to the Semantic Web, there are restrictions to the definition above. This is because knowledge representation is only used to describe content and formal aspects of web resources. The resulting description is expressed by a specific markup language for metadata: Resource Description Format (RDF). This aspect of the representation of web resources described by RDF is the most evident layer, but within this aspect there are also other less visible levels. Each one uses application tools based on conceptual schemes and logic tools. All of this is expressed by a declarative knowledge representation methodology. The graphic below (Figure 1) shows the logics, the tools and the levels of knowledge representation involved in the Semantic Web.

This figure will be explained step by step.

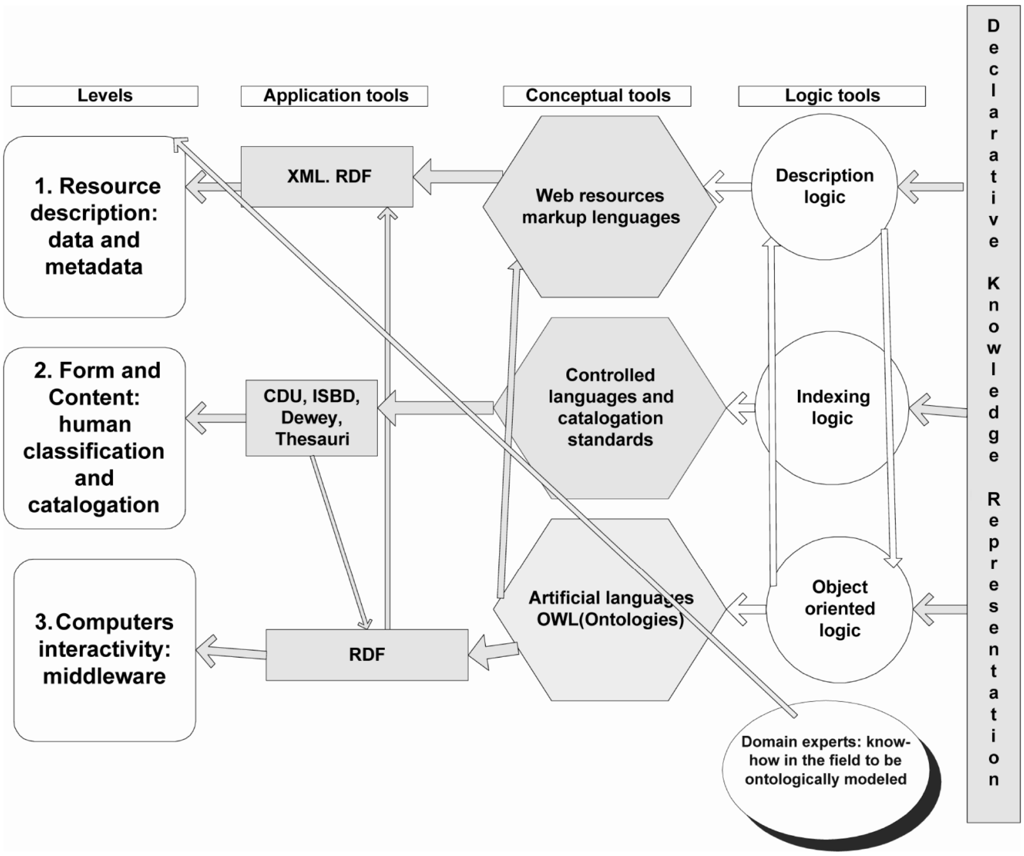

Figure 1. Semantic Web: logics, tools and levels of knowledge representation

2.1 First level: methodologies for knowledge representation in intelligent systems: from the Procedural form to the Declarative form

Figure 1 shows that the most abstract level of representation in the Semantic Web is the declarative knowledge representation. Historically, intelligent systems have used two methodologies: the Procedural form and the Declarative form. In the former, knowledge is integrated in the computer program. This methodology has many advantages because of its high level of specificity where algorithms are tailor-made. On the other hand, it does lack versatility, and making changes to the program is difficult. The procedural methodology has been in use for a long time, but since the 80s the declarative knowledge representation has become more widespread. In this methodology the representation is independent from the computational process. It is flexible and has a strong logical base. However, its great level of abstraction may result in a lack of consistency in maintaining its logic.

In the Semantic Web the markup language that is expressed by the metadata to describe web resources has its genesis in the procedural methodology. The Standard Generalized Markup Language (SGML) was developed by Charles Goldfarb in 1969. Originally called General Markup Language, it was used to exchange documents at IBM. It was quickly adopted for document circulation by the US Department of Defense and by the Office for Official Publications of the European Communities, both of them IBM clients. In 1986 it was adopted by the International Standards Organization. Thus the GML became the widely used ISO 8879 standard (Bryan 1998).

This peculiar situation shows two aspects of SGML: it was created for corporate documents and it was meant for data processing, specifically for the exchange of data. Hence, the key words for ISO 8879 are: "Data processing, documentation, Logical structure, programming (computers), artificial languages, programming languages (ISO 1986)." It should be noted that the success of SGML is not only due to its data processing aspects, but also to its capability in exchanging specific information in a semantically expressive non-procedural mode.

When SGML evolved into Extensible Markup Language (XML), it was able to apply the Description or Terminological Logics, which is a sort of declarative representation (Lambrix 2005):

> Description logics are knowledge representation languages tailored for expressing knowledge about concepts and concept hierarchies … They are considered an important formalism unifying and giving a logical basis to the well known traditions of frame-based systems, semantic networks and KL-ONE-like languages, object-oriented representations, semantic data models, and type systems. The basic building blocks are concepts, roles and individuals. Concepts describe the common properties of a collection of individuals and can be considered as unary predicates which are interpreted as sets of objects. Roles are interpreted as binary relations between objects.

RDF is a specific application of XML and it is the main tool for the Semantic Web. Its importance arises from its blending procedural specificity with the potential of declarative abstraction by using description logic (see Figure 2).

The result is a powerful and expressive format that allows knowledge representation in the digital world (http://www.w3.org/RDF/):

> The Resource Description Framework (RDF) integrates a variety of applications from library catalogs and world-wide directories to syndication and aggregation of news, software, and content to personal collections of music, photos, and events using XML as an interchange

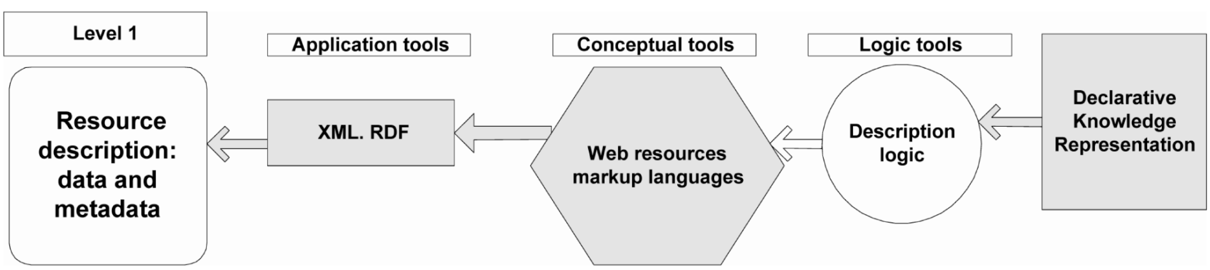

Figure 2. Level 1 for resource description

syntax. The RDF specifications provide a lightweight ontology system to support the exchange of knowledge on the Web.

Consequently, through the description logic combined with a markup language, RDF becomes a controlled language in which the syntax is independent from system procedures and the terminology is adequate to the application domain. It has well-defined and very expressive semantics.

2.2 *Second level: the logic of indexation for cataloguing and classifying*

Recent studies say that XML is the *lingua franca* of the Web, as RDF is the area of work of IT professionals and web designers. However, XML is not exactly a *lingua* or language. It is actually a set of syntax rules, and therefore an annotation. In order to create a language based on XML syntax it is necessary to have the means for providing this annotation of semantics. XML follows the descriptive logic indicated above. It organizes the representation of system domains by concept hierarchies ordered into classes and subclasses of digital objects. This kind of organization is similar to the controlled languages used in library subject catalogs. It is therefore possible to take advantage of the experience of Library and Information Science (LIS) professionals in classifying and cataloguing (see Figure 3).

Nevertheless, there is a problem. Even though the methods for building controlled languages for document management are similar to the hierarchical structure of XML syntax, the application is different. As mentioned above, the goal of the Semantic Web is to represent Web resources. This means it describes digital objects and not references alone like library systems do. It should be noted that it is not the same to index an object and to index something that is the reference to that object. The main focus of LIS is to reference formal and content aspects of documents. The behavior of documents in a digital environment is limited to storage, search and retrieval. Hence, other activities like creation, revision or modification are accomplished by other systems, such as Content Management Systems (CMS).

It is important to integrate all the functions of documents with different systems. But on the other hand, digital objects have two different aspects: the data and the metadata. The metadata structure is similar to the reference of a document catalog, with additional features. For example, the Dublin Core Metadata Initiative (DCMI) has many different core elements:

- Content elements such as title, keywords, abstract, source, language, spatial and temporary coverage;
- Relationships with other documents;
- Copyright and authoring elements such as creator, contributor, publisher; and
- Elements that relate to the instance of the document: date, resource-type, format, resource-identifier.

Many of these DCMI elements are analogous to the descriptions in classification and cataloguing systems. When considering documents and other digital objects in an information system domain, there are some aspects of digital objects such as attributes, behavior, relations and cardinality, that are expressed by another logic. Nowadays the leading point of view to express these aspects is the object-oriented one.

2.3 *Third level: the logic of the object oriented paradigm*

The transformation from procedural form to declarative form has brought important changes to the development of programs. The procedural paradigm considers the computer program as a collection of

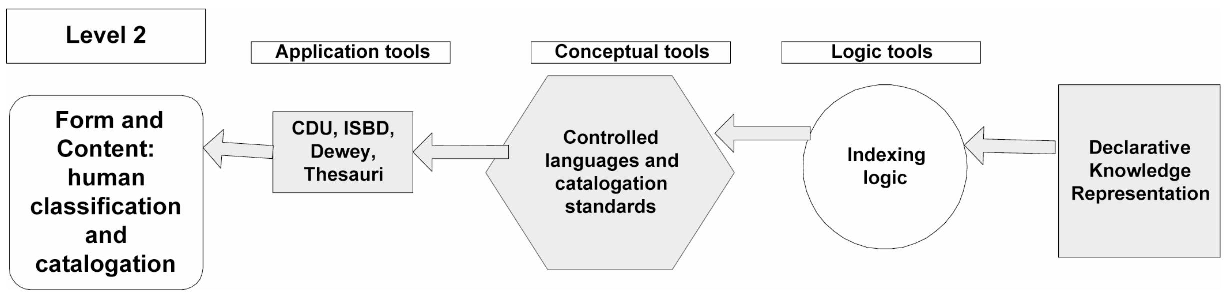

Figure 3. Level 2 for form and content

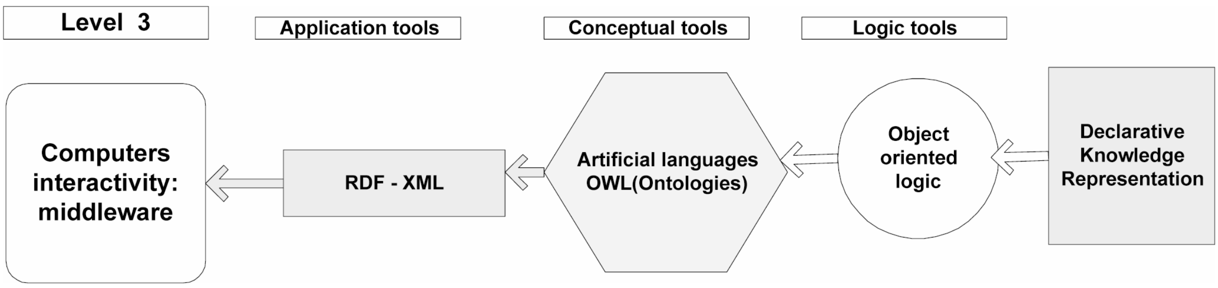

Figure 4. Level 3 for computer interactivity

functions or a list of instructions given to the computer. The Object Oriented Paradigm clashes with this traditional point of view, as it considers a computer program a set of interacting individual units or objects, where any object can manage its own state and operations (see Figure 4).

This Object Oriented logic makes domain assumptions explicit in a very understandable way for humans. But those domain assumptions still need to be complex enough to be used by the system. This balance is possible by separating the domain knowledge from the operational knowledge by means of terms used for representing concepts that are an abstraction of the objects' main properties.

The following is a brief overview of the main features of the object oriented logic: *Classification* is the capability to unify definition of data and behavior in a hierarchical structure of elements that belong to a domain. The classes are instances or *Objects*, which define each specific part of a system. These objects are explicit abilities: *Inheritance, Encapsulation, Polymorphism, Abstraction*. Inheritance is also called generalization because it is a hierarchical relationship between classes of objects. Encapsulation is a type of isolation applied to the data, which ensures that an object can be changed only through established channels or methods. Polymorphism is the ability of objects to react differently to different information – that is, two or more classes can respond differently to the same message. All of this would be of no use without abstraction, which is the ability of the objects to disregard the details of an object's sub-class and work at a further generic level when suitable (Rumbaugh et al. 1991). The object oriented logic approach is very intuitive. It allows a direct mapping of objects from the real world. It is, however, difficult to build a large, coherent and complete representation of the system domain using a hand-made object hierarchy.

It must be acknowledged that currently many experiences are being carried out applying alternative types of logics, such as *fuzzy* logic and *multiple-valued* logic for Semantic Web development, both in databases and other repositories of documented information. Nevertheless, the development of ontologies using the most common standards still focuses mostly on the object oriented paradigm, which actually may complement and expand itself through the usage of frame-based representation. Such is not the subject of this work (Lassila and McGuinness 2001).

3. Levels of representation and interactions in the construction of ontologies

The guide for ontology building with the graphic tool Protégé from Stanford University considers that object-oriented modeling and ontology engineering involve many common steps:

- An iterative process;
- the division of concepts of the domain into classes;
- the arrangement of the concepts in a hierarchy;
- the process of specifying which attributes property classes can have and indicating constraints on their values; and,
- the filling in of slot values.

The same authors indicate that the logic is the same. However, an ontology reflects the structure of the world and it is often about structure of concepts. Consequently, the actual physical representation is not an issue. An object-oriented model reflects the structure of the data and is usually about behavior. Therefore, it describes the physical representation of data, e.g., long, int., char, etc. (Noy and McGuinness 2001). When working with professionals from different backgrounds in a multidisciplinary group, it is often considered necessary to remember that sometimes we are observing the real world and sometimes we are representing the data of the domain for the information system.

Now both the object-oriented logic for the construction of ontologies and the object-oriented paradigm will be analyzed within the parameters of their relation with the methods of documentary classification. As mentioned above, one of the great differences between documentary classifications and ontologies is their aim. The former intends to reference an object. The latter achieves a physical and semantic description of the object in a specific domain of computer system application.

This important dissimilarity is evident also in the development of classifications in LIS. In fact, according to Gnoli and Poli (2004), historically there have been two approaches to the description of objects and events: the epistemological one and the ontological one. In the epistemological orientation there is a previous and external theoretical structure based on the class hierarchy of scientific disciplines, e.g. the *Dewey Decimal Classification* (*DDC*). In the ontological line the description focuses on features inherent to events and objects, which can be considered within the *Colon Classification*, the *Bliss Classification*, and in general in all faceted classifications.

they have taken advantage of the oldest existing experiences in the field of LIS. The classifications that use an ontological orientation can be a precious tool and a framework for the development of ontologies. This is the standpoint of the ILC project of ISKO, Italy. For further reference, please refer to http://www.iskoi.org/ilc.

4. Conclusions

Some time ago Tom Gruber (2004, 8), one of the pioneers in the development of ontologies, said: "Every ontology is a treaty – a social agreement – among people with some common motive in sharing." The collaborative approach to the construction of ontologies offers an excellent opportunity to integrate different levels and the disciplines involved. Therefore, it is of importance to bear in mind the types of reasoning and the levels of knowledge representation. In order to make this collaboration successful, it is essential to harmonize all aspects of the work, by providing a space for each discipline and having an environment which hinges on respect for all the fields involved (see Figure 5).

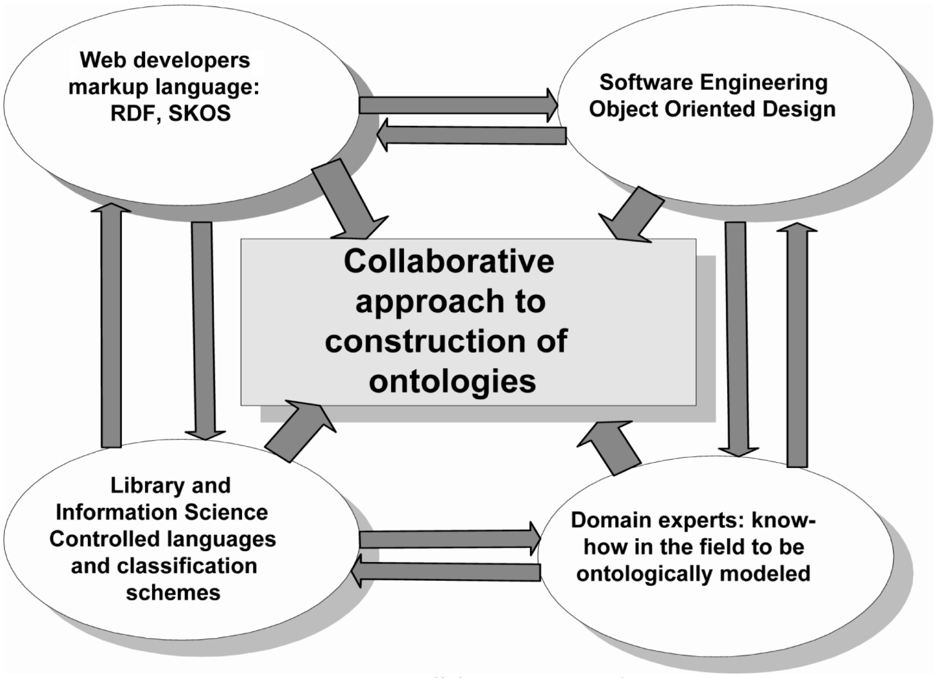

Figure 5. Collaborative approach

The Semantic Web ontologies did not inherit the tradition of bibliographical classifications. They arose within the tradition of artificial intelligence. In fact, the objective of Semantic Web ontologies is not a broad-spectrum of knowledge but the restricted representation of application domain. Gradually,

The communication between collaborators from different disciplines is difficult and cost analysis in such projects is complex, for ontology building, maintenance and reuse are time consuming activities. There are various research trends related to these aspects such as:

- Methodologies for collaboratively creating and managing shared information and modeling semantically heterogeneous data sources and services (Hodgson 2005).
- Semantic community support systems and collaboration applications, such as Groupware tools for supporting collaborative ontology design (Díaz 2005).
- Cost Estimation Models for Ontology Engineering (Institut für Informatik, Freie Universität Berlin 2005).

It is evident that much remains to be done. It will be easier if we work together.

References

Bosch, Mela. 2001. Documentos y lenguaje de marcado: conceptos, problemas y tendencias (Documents and markup language: concepts, problems and trends). *El profesional de la información* 10: 4-9.

Bryan, Martin. 1998. *SGML an author guide*. New York: Addison Wesley.

Davis, Randall, Shrobe, Howard, and Szolovits, Peter. 1993. What is a knowledge representation? *AI magazine* 14:17-33. http://groups.csail.mit.edu/medg/ftp/psz/k-rep.html

Díaz, Alicia. 2005. CO-Protégé: A Groupware Tool for Supporting Collaborative Ontology Design with Divergence. Lifia, Fac. Informática- UNLP, La Plata, Argentina- Loria, Campus Scientifique, Vandœuvre-lès-Nancy cedex, France. http://protege.stanford.edu/conference/2005/slides/6.2_A.Diaz_Co-Protege_slices_and_flyer.pdf.

Gnoli, Claudio and Poli, Roberto. 2004. Levels of reality and levels of representation. *Knowledge organization* 31: 151-60.

Gruber, Tom. 2004. Interview. In: http://www.sigsemis.org/bulletins/sigsemis1(34).pdf

Hodgson, Ralph and Keller, Paul. 2005. Collaborative ontology-based systems: innovator perspectives and demonstrations of new open standards and technologies in support of ontology engineered solutions: TopQuadrant and NASA Ames. *Collaborative Expedition Workshop #38, at NSF Semantic Conflict, Mapping, and Enablement: Making Commitments Together*. http://www.topquadrant.com/documents/talks/TQ%20Ontology-Based%20Collaborative%20Environments%20(v4).pdfBased%20Collaborative%20Environments%20(v4).pdf.

Institut für Informatik, Freie Universität Berlin. 2005. *Ontology engineering cost estimation with ONTOCOM*. http://ontocom.ag-nbi.de//index.html.

ISO 8879. 1986. Information processing. Text and office systems. *Standard Generalized Markup Language (SGML)*. JTC 1/SC 34; ISO Standards 35.240.30.

Lambrix, Patrick. 2003. *Description logic*. Intelligent Information Systems Laboratory, Department of Computer and Information Science, Linköping University, S-581 83 Linköping, Sweden: http://www.ida.liu.se/labs/iislab/people/patla/DL/index.html.

Lassila, Ora and McGuinness, Deborah. 2001. *The role of frame-based representation on the Semantic Web*. Software Technology Laboratory, Nokia Research Center Knowledge Systems Laboratory, Stanford University. http://www.ida.liu.se/ext/epa/ej/etai/2001/018/01018-etaibody.pdf.

Noy, Natalya F. and McGuinness, Deborah. 2001. *Ontology development 101: a guide to creating your first ontology*. Stanford University: http://protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html.

Rumbaugh, James, Blaha, Michael, Premerlani, William, Eddy, Frederick, and Lorensen, William. 1991. *Object oriented modeling and design*. Englewood Cliffs, NJ: Prentice Hall.