Semantic Web Domain Knowledge Representation Using Software Engineering Modeling Technique

Minal Bhise

DAIICT, Gandhinagar, Gujarat, India 382007

minal_bhise@daiict.ac.in

Abstract. The semantic web offers a great deal of deviation from the way in which the current search engines which are based on the traditional information search theory work. Semantic search is carried out by ontology based intelligent information retrieval. So a good semantic search needs a good ontology. The ontology developers need more familiar notations and tools for a uniform representation of ontologies. UML being a standard modeling language in software engineering, it is better supported in terms of expertise and the tools as compared to the upcoming semantic web languages. This work proposes a representation technique which is based on software engineering standards namely UML for modeling domain knowledge of the Semantic Web. The ontology for Company Domain has been presented using this software engineering modeling technique. The UML tool like Rational Rose tool can be used to provide support for modeling complex ontologies of the given domain.

Keywords: semantic web, ontology, UML, domain knowledge modeling, Ontology Representation Language ORL

1 Introduction

The semantic web will let the computer software relieve us of much of the burden of locating resources on the web that are relevant to our needs and let the system access and use the information more efficiently. The semantic web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation (Berners-Lee,Hendler & Lassila, 2001)**.** Facilities to put machine-understandable data on the web are becoming a high priority for many communities. The web can reach its full potential only if it becomes a place where data can be shared and processed by automated tools as well as by people. To enable automation, integration and reuse of data resources on the web need to be encoded in structured machine-readable descriptions of their contents expressed using terms defined in domain ontology. Ontologies are set of knowledge terms, including the vocabulary, the semantic interconnections, and some simple rules of inference and logic of some particular topic (Thomas & Gruber, 1993)**.** In order to create, interpret and compare meta-data annotations, ontologies, explicit definitions of the vocabulary used in an information source, are needed. While meta-data can often be generated from the content of an information source, interpretation and comparison of this data needs the machine to understand these different, new and the existing ontologies.

The various Ontology Representation Languages (ORLs) like KIF, SHOE, Topic Maps, DAML, OIL and OWL lack good visual modeling tools which are a must for human comprehension of ontologies (Cranefield, 2001). Further these visual models should have an ability to map to machine understandable representations which will be used by the agents for inferencing and integration. While ontologies are built on the Artificial Intelligence concepts, the ontology building tools are not. The techniques used so far for knowledge representation are based on Knowledge Interchange Format KIF which has very small following and that too within AI community only (Genesereth & Fikes, 1994). The software experts need more familiar notations and tools for a uniform representation of ontologies. UML being a standard modeling language in software engineering, it is better supported in terms of expertise and the tools as compared to the upcoming semantic web languages. The UML is expressive and standardized modeling language which has large user community and very good commercial tool support in the form of IBM Rational Rose, MagicDraw, JUDE and ArgoUML (UML Tool Survey, 2005)**.** The use of UML for ontology representation will allow many mature UML tools, models and expertise to be applied to knowledge representation systems not only for visualizing the complex ontologies but also for managing the ontology development process. The UML is a graphical modeling language hence very easy for human comprehension. The UML is designed for model building by human experts, while ORLs are designed to be used at run time to help intelligent processing methods. But there are certain differences in the languages. One has to identify the corresponding elements in the two languages and also need to make sure that translations are backed by the semantics of the languages. The UML class diagrams provide a static modeling capability that is integral part of ontology representation (Cranefield & Purvis, 1999). UML Object Diagrams represent instances of the Class Diagram hence can represent the instances of the concepts represented by Class Diagram. Semantic Web Technology has been built on the object-oriented paradigm so it is better to use the same paradigm for modeling and representing knowledge (Cranefield 2001).

This work demonstrates the use of UML to model ontology for the Company Domain. An Overview of Technology has been presented in Section 2. The Section 3 represents the application of the UML for representing ontology for the Company Domain. The Section 4 contains Discussions and Future Work, and the last section, Section 5 presents Conclusions.

2 Overview of the Technology

The technology supporting domain knowledge representation for semantic web consists of UML domain models, semantic web pillars namely ontology, agents and markup language and the mapping for UML models to ORL representations.

2.1 UML Domain Models

A class diagram gives static view of the structure of the system. A class diagram presents set of classes, interfaces and collaborations and their relationships. The class diagrams are often used to model vocabulary and collaborations of the system. The details about the notation are available elsewhere (Booch, Rum Baugh & Jacobson, 2004). The class diagram can show association, aggregation and inheritance type of relationships. UML Object Diagrams can depict the instances of Class Diagram and hence help in representing knowledge about the domain. It can be represented as instances of the concepts which are depicted as classes in the class diagram. The relationships in the object diagrams are instances of the associations (and other types of relationships) among concepts in the class diagram.

2.2 Semantic Web and Domain Knowledge Representation

Each of the semantic web pillars is related to the domain knowledge representation in some way. Together they form an intelligent information retrieval system which is ontology based.

2.2.1 Ontology

Ontology is a set of logical axioms designed to account for the intended meaning of a vocabulary. Ontology specifies the meanings of the terms in a given vocabulary by explicitly defining the relationships between the terms. By defining shared and common domain information, ontologies can help people and machines communicate concisely supporting semantics exchange, not just syntax (Berners-Lee, Hendler & Lassila, 2001). The ontology building process needs standard interfaces and tools for creating and updating these ontologies. The software engineering based modeling technique discussed in this work addresses this problem.

2.2.2 Markup Languages

A markup language is a tool for adding information to the documents. Semantic markup is expected to have universal expressive power, syntactic and semantic interoperability. The XML (eXtensible Markup Language) does not provide semantic interoperability. Using XML as a base , a number of new markup languages have been developed to meet semantic interoperability, these are RDF(Resource Description Framework), RDFS, SHOE, DAML+OIL and recently OWL (Daconta Michael, 2003). The work done earlier in domain knowledge representation using software engineering modeling techniques was concentrated towards developing the mapping of the software engineering models to the RDF, RDFS and other markup languages (Cranefield, 2001). At present the researchers are trying to map the domain models to the OWL representation (Djuric, Gasevic & Devedzic, 2004).

2.2.3 Intelligent Agents

Ontologies and markup languages will make the semantics of the documents available to the machines. Software agents will make use of the semantic content, actually interpreting and integrating the content of documents (and queries) to perform tasks for users. Agents will have to carry out the task of finding out and using the information on the web from several resources on their own, processing them and integrating the results and presenting them to the users or carrying out inferencing based on those results. So the different ontologies should be able to work together in order to make all this possible (Hendler, 1999). Therefore we need a standard way of representing these ontologies which are understandable by human as well as agents for automatic processing. For the reasons discussed in Section 2, the UML is the obvious choice.

2.3 UML to ORL Mapping

The UML class diagrams represent the important concepts from the problem domain in the form of classes and their attributes and methods. The relationships among these concepts are represented by relationships among these classes. One can also express the cardinality and other types of constraints using the UML notation available (Booch, Rum Baugh & Jacobson, 2004). The ontologies define concepts from the problem domain and relationships among them. The XML Model Interchange Language defines a standard way to serialize the UML diagrams (Cranefield, 2001). So the knowledge expressed in the form of UML diagrams can be directly comprehended by human because of its standard graphical representation as well as by ontology editors. There are also a number of Java class libraries available to provide an interface to various applications accessing this information. The UML diagrams also can be accessible and processed by computers because of XMI and associated libraries or APIs defined by MOF (Baclawski, Kokar, Kogut, Hart, Smith, Holmes, Letkowski & Aronson, 2001). The XMI specifies how a model stored in a MOF-based model repository can be represented as an XML document. The UML class diagrams can be mapped to RDF schemas (Falkovych 2003). UML classes can also be mapped to sets of Java classes. These classes correspond to the classes in the class diagram.

The domain experts create an ontology using ontology editor and the graphical representation of ontology using UML tool like Rational Rose. The XMI files are created. A pair of XSLT eXtensible Stylesheet Language Transformation then creates java files and ORL representation of ontology respectively. The java classes can be used by the applications for representing knowledge as in memory data structures. The ORL representation can be used for domain specific information.

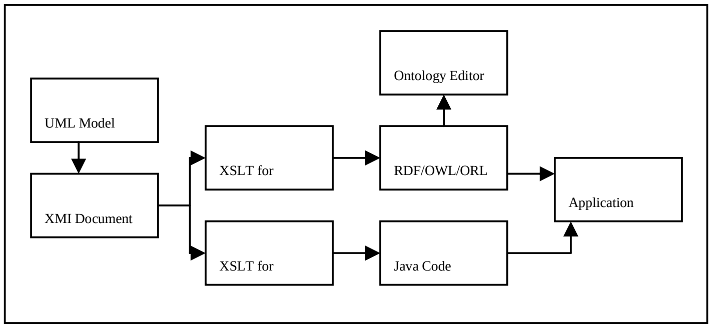

Figure 1: Schematic for Domain Model Mapping

Presently the RDF transformation of XMI files is available and not much work has been done for XMI to other ORL mapping. There is some work done for direct XMI representation of UML model mapping to OWL (Djuric, Gasevic & Devedzic 2004). The schematic for UML model to ORL and Java mapping has been shown in Figure1.

3 The Company Ontology in UML

The Company has name, director, turnover and head office. The Company consists of Concepts like Office and Department. These are depicted as aggregation relation among concepts Department, Office and Company. The various departments like R&D and Sales are related to Department concept through inheritance relationship. The ontology for the company is shown in Figure 2 below using notation described elsewhere (Bruijn & Krummenacher, 2005).

Concept Company
hasname oftype name
hasdirector oftype name
hasturnover oftype number
hasheadoffice oftype string
hasConcept Office
hasname oftype name
hasaddress oftype string
hasConcept Department
hasname oftype name
hasbudget oftype number
Concept R&D Subconcept of Department
haspatents oftype string
hasprojects oftype string
Concept Sales Subconcept of Department
hasproduct oftype string
hassale oftype number

Figure 2: Domain Ontology for Company Domain

A portion of the class diagram depicting this ontology has been represented in Figure 3 using UML. The relationship among Company, Department and Office has been depicted as aggregation relation among classes Department, Office and Company in the class diagram. The Office has name and location. The department has name and budget. The various classes like R&D and Sales departments are related to Department class through inheritance relationship.

Schematic showing mapping of UML diagrams to Java and OWL files which can be used by agents handling corresponding semantic web applications has already been depicted in Figure1.

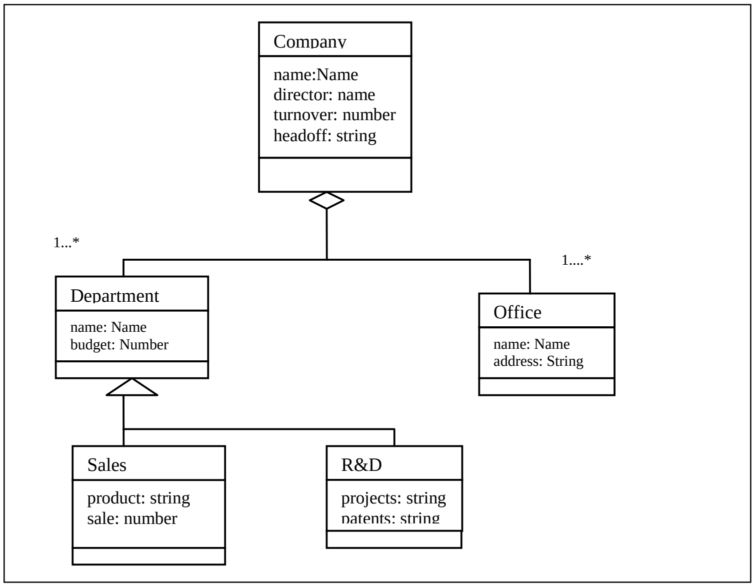

Figure 3: UML Class Diagram for Company Domain Ontology

4 Discussion and Future Work

The major problem faced by semantic web ontology developers is the availability of visual models for ontology representation. This problem has been addressed in this work through the use of software engineering based UML class diagram. Some issues still need to be solved. Using XML encoding of both XMI and OWL one can create an XSLT file that maps the two models (Djuric et.al., 2004, Djuric, Gasevic & Devedzic, 2005). But XSLT is very cumbersome when used for complex ontologies. Further, UML associations are not first class concepts as they need defined end points but properties in ontologies are first class objects. To reconcile this difference, a modest extension to the UML infrastructure has been proposed. The metamodel MOF (Meta-Object Facility) has been introduced (Baclawski et. al., 2001, Baclawski, Kokar & Aronson, 2001). The architecture followed by this meta-metamodel is called as Model Driven Architecture MDA. The MDA divorces implementation details from business functions. Thus it is not necessary to repeat the process of modeling an application or system functionality and behaviour each time a new technology comes along. Also it is driving UML more and more formal, so that is useful when it comes to using it for the automatic inferencing by agents. A complete MDA specification consists of a platform independent model like UML plus one or more PSMs (Platform-Specific Models) (Anneke, 2003). The problem of transformation between ontology and MDA-based languages is solved using XSLT. But still no commercial MDA tools are available which can process the models at M3 and M2 layers. The existing UML tools support the models till M1 layer very well (Djuric et. al., 2004).

Most of the work done so far is centered on the use of class diagram for ontology representation. But UML is much richer beyond class diagram. Can other diagrams be used for looking at other aspects of the ontologies? I understand that different knowledge aspects of the agent based intelligent systems can be looked at using the other UML diagrams as well. Very less work has been done so far in this direction. A proposal exists for modeling DAML-S using other UML diagrams (Falkovych, 2003).

5 Conclusion

Ontologies of semantic web play a major role when information is accessed by agents handling the applications as described in section 1. The major problem faced by ontology developers is the availability of standard and easy to use tools for visualizing the ontologies while building them. This work addresses this problem by using OMG's UML for representing domain ontologies. The mapping of UML models to Java and OWL representations to be used by agents handling corresponding semantic web applications is possible. A case study representing ontology of company domain has been presented. Some issues still need to be solved as discussed in the previous section.

References

[1] Anneke K. (2003). *MDA Explained*. Pearson Education.

[2] Baclawski K., Kokar M., Kogut P., Hart L., Smith J., Holmes W., Letkowski J., Aronson M. (2001) Extending UML to support Ontology Engineering for the Semantic Web, In the Proceedings of UML 2001, Toronto, Canada, October 1-5, 2001.

[3] Baclawski K., Kokar M., Kogut & Aronson M. Metamodeling Facilities, working paper presented at OMG Technical Meeting, Danvers, MA, USA, July 9-13, 2001.

[4] Berners-Lee T., Hendler J., Lassila O. (2001). The Semantic Web. *Scientific American*, 284, 35-43.

[5] Booch G., Rumbaugh J, Jacobson I. (2004). *The Unified Modeling Language User Guide*. Pearson Education.

[6] Bruijn J.D., Krummenacher R. (2005). *Technical Report on Ontology Representation Language*, DIP

[7] Cranefield, S. UML and the Semantic Web, In Proceedings of the International Semantic Web Working Symposium SWWS, Palo Alto, July 30- August 1, 2001.

[8] Cranefield S. and Purvis M. UML as an Ontology Modeling Language. (1999). In the proceedings of workshop on Intelligent Information Integration, 16th International Joint Conference on Artificial Intelligence (IJCAI-99).

[9] Daconta, Michael C. (2003). *The Semantic Web*. Wiley Publishing.

[10] Djuric D., Gasevic D., Devedzic D. Converting UML to OWL Ontologies. WWW 2004, May 17-22, 2004, NY, USA

[11] Djuric D.,Gasevic D.,Devedzic D.(2005). MDA for Ontology Modeling. *Journal of Object Technology*, 4(1), 109-128.

[12] Falkovych K. (2003). UML for the Semantic Web: Transformation-Based Approaches. In *Knowledge Transformation for the Semantic Web, Frontiers in Artificial Intelligence & Applications*. (pp 92-106). IOS press.

[13] Genesereth M.,Fikes R.(1994). *The Knowledge Interchange Format (KIF) Reference Manual.* Stanford University, USA.

[14] James H. (1999). Is there an Intelligent Agent in your Future, *Nature*, 121-129.

[15] Thomas R. Gruber.A. (1993). Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*, 5(2),199-220.

[16] The UML Tool Survey (2005).

http://www.objectsbydesign.com/tools/umltools_byCompany.html