Constructing Ontologies in OWL using Protégé-2000

Monika Singh[1], Sanjay Kumar Malik [2]

[1]M.Tech.-IT, IV Sem, University School of Information Technology, GGS Indraprastha University, Delhi
[2] University School of Information Technology, GGS Indraprastha University, Delhi
singh.monika@gmail.com[1], sdmalik@hotmail.com[2]

Abstract - In the light of improving the World Wide Web, researchers are working towards the Semantic Web, a Web of information that machines can understand and process. Ontologies and ontology-based applications are its basic ingredients. Protégé-2000 is an ontology-development and knowledge-acquisition environment developed in Stanford Medical Informatic. It has a graphical user interface which enables ontology developers to concentrate on conceptual modeling without knowing or thinking about syntax of an output language. Protégé-2000 has a flexible knowledge model and extensible plug-in architecture. This paper describes how Protégé-2000 can be used for constructing ontologies in OWL. We show an ontology building process that is the basis for an application development. It will start with the brief description of semantic web, ontology and ontology languages. OWL is the W3C standard ontology language for the Semantic Web and Protégé-OWL the most widely used editing tool for OWL.

KEYWORDS

Semantic Web, Ontologies, Protégé, design methodologies, OWL

1. INTRODUCTION

In the light of improving the World Wide Web, researchers have found enormous amount of information and knowledge. An emerging problem of significant importance is the efficient retrieval and reuses these resources.

A promising approach proposed by Tim Berners-Lee is the reformation of the Web, as it exists, into the "Semantic Web". As Berners-Lee says "The Semantic Web is an extension of the current Web in which information is given well-defined meaning [1]". It is the idea of having data on the Web defined and linked in a way that it can be used for more effective discovery, automation, integration, and reuse across various applications.

Ontologies are becoming the corner stone of the Semantic Web (SW).Ontologies aim at capturing domain knowledge in a generic way and provide a commonly agreed understanding of a domain. They are shared conceptualizations of a domain, and they possibly include the representations of these conceptualizations [2]. Ontologies are independent from the applications that use them. This leads to easier software and knowledge maintenance, and contributes to the semantic interoperability between applications [3].

Among the representation formalisms for ontologies, the Web Ontology Language (OWL) is the widely accepted standard for representing and sharing knowledge in the Semantic Web context.

The OWL language is divided into three syntax classes:

- OWL-Lite - classification hierarchy and simple constraints
- OWL-DL - maximum expressiveness with desirable computational properties for reasoning.
- OWL-Full - maximum expressiveness and syntactic freedom of RDF with no computational guarantees.

Particularly, OWL-Lite and OWL-DL belong to the description logics [4]

From the existing tools (e.g. WebOnto, OntoEdit), Protege-2000 is chosen for implementation because it enables the construction of domain ontologies, customized data entry forms to enter data. Protégé allows the definition of classes, class hierarchies, variables, variable-value restrictions, and the relationships between classes and the properties of these relationships [16].

This paper focuses on how Protégé-2000 can be used for constructing ontologies in OWL. The approach is illustrated through simple examples of creating ontology.

2. PROTÉGÉ-2000

Protégé is an ontology and knowledge base editor produced by Stanford University. Protégé is a tool that enables the construction of domain ontologies, customized data entry forms to enter data. Protégé allows the definition of classes, class hierarchies, variables, variable-value restrictions, and the relationships between classes and the properties of these relationships. Protégé is free and can be downloaded from http://protégé.stanford.edu. Protégé comes with visualization packages such as OntoViz, EZPal, etc.; all of these help the user visualize ontologies with the help of diagrams. Stanford University is doing a magnificent job of continually improving Protégé. As part of its last update, Protégé now includes an interface for SWRL (Semantic Web Rule Language), which sits on top of OWL to do math, temporal reasoning, and adds Prolog-type reasoning rules. Stanford has a tutorial that covers the basics of using Protégé with the OWL plug-in. Additional support can be obtained by consulting others on the Protégé/OWL news group.

The main strong point of Protégé is that it supports at the same time tool builders, knowledge engineers and domain specialists. This is the main difference with existing tools, which are typically targeted at the knowledge engineer and lack flexibility for meta-modeling. This latter feature makes it easier to adapt Protégé to new requirements and/or changes in the model structure.

3. ONTOLOGY DEVELOPMENT

We describe an iterative approach to ontology development: we start with a rough first pass at the ontology [5]. We then revise and refine the evolving ontology and fill in the details we start ontology development with collecting requirements for the envisaged ontology. By nature this task is performed by a team of experts for the domain accompanied by experts for modeling. The outcome of this phase is (i) a document that contains all relevant requirement specifications (domain and goal of the ontology, design guidelines, available knowledge sources, potential users and use cases and applications supported by the ontology) (ii) a semi-formal ontology description, i.e. a graph of named nodes and (un-)named, (un-)directed edges, both of which may be linked with further descriptive text.

To operationalize a methodology it is desirable to have a tool that reflects and supports all steps of the methodology and guides users step by step through the ontology engineering process. Along with the development of the methodology we therefore extended the core functionalities of Protégé.

First, we would like to emphasize some fundamental rules in ontology design to which we will refer many times. These rules may seem rather dogmatic. They can help, however, to make design decisions in many cases. . Further, full support for the latest ontology languages is lacking.

Ontology is typically built in more-or-less the following manner:

1. **Acquire domain knowledge**

Assemble appropriate information resources, their characteristics and the terms to describe things in the domain of interest. The output is a natural-language ontology specification document.

2. **Organize the ontology**

Domain terms are identified as concepts, instances, verbs relations or properties and each are represented using an applicable informal representation.

3. **Flesh out the ontology**

Add concepts, relations, and individuals to the level of detail necessary to satisfy the purposes of the ontology.

4. **Check and classify your work**

Reconcile syntactic, logical, and semantic inconsistencies among the ontology elements. Consistency checking may also involve automatic classification that defines new concepts based on individual properties and class relationships.

5. **Commit the ontology**

Final verification of the ontology by domain experts and the subsequent commitment of the ontology by publishing it within its intended deployment environment.

4. METHODOLOGY FOR ONTOLOGY CONSTRUCTION

At present the construction of ontologies is very much an art rather than a science [6].This situation needs to be changed, and will be changed only through an understanding of how to go about constructing ontologies. In short what is needed is a good methodology for developing ontologies. This attempt to formalize the ad-hoc process consists of the following steps [7]

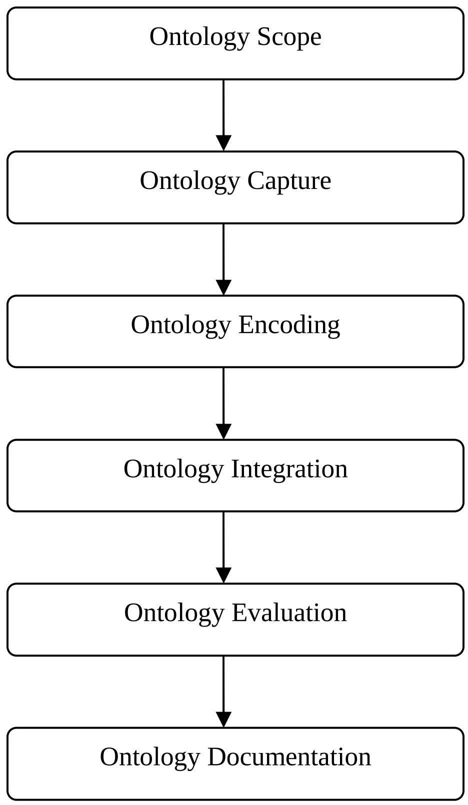

Fig 1. Ontology Construction Methodology

Scope
Identify the range of intended users, determine the purpose of the ontology, and identify user requirements for systems using ontology.

Capture
Identify key concepts and relationships in the domain of interest, produce precise definitions for such concepts and relationships.

Encoding
Committing to the basic terms that will be used to specify the ontology, choose representation language and coding the ontology[8].

Integration

Use existing ontologies, task is non-trivial, identify synonyms in a given ontology and extend it where no suitable concept exists.

Evaluation
Checking ontology against purpose, user requirements and competency questions (CQ).Ontology must be able to answer all the given CQ.

Documentation
Effective knowledge sharing requires adequate documentation, all assumptions should be documented.

5. SIMPLE EXAMPLE OF CREATING ONTOLOGY

Look at the Protégé documentation for the details on how to create class, subclasses and slots.

Employee Ontology
In a project organization, the Employee Ontology [9] is defined as the conceptualization of the Employee who has an Employee Position in the organization and is identified by an Employee Name as well as has Responsibilities which include some Projects [13].

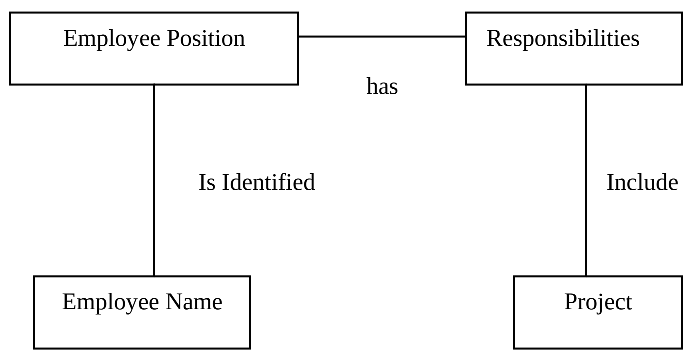

Fig 2. Employee Ontology

We present the Employee Ontology as the combination of the ontology name and a tuple where the elements of the tuple can be complex elements as defined below[10]:
Employee [Employee Position, Employee Name and Responsibilities] where

- Unique identification of Employee
  Employee Position.
- Unique identification of Employee Position
  Employee Name
- Aggregation
  Responsibilities

6. CONCLUSIONS AND FURTHER RESEARCH

It is quite clear Ontology development is necessarily an iterative process. Among several viable alternatives, we will need to determine which one would work better for the projected task, be more intuitive, more extensible, and more maintainable. We also need to remember that an ontology is a model of reality of the world and the concepts in the ontology must reflect this reality.. We have described a tool-assisted method for building the basis for ontologies adopted from domain analysis. The ontologies built by this method may not yet be comprehensive or formal enough for some purposes but they provide sufficient information and concepts to facilitate the task of ontology coding and formal documentation. The hardest part is designing a good ontology before implementing it with Protégé. Working with Protégé to generate OWL is much easier and faster than conventional programming. Once you have a design, implementing it in Protégé only takes hours. The goal is to create learning ontologies that respond to changes in the environment.

REFERENCES

[1] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. Scientific American, May 2001.

[2] Gruber R. Formal Ontology in Conceptual Analysis and Knowledge Representation. Chapter "Towards principles for the design of ontologies used for knowledge sharing"in Conceptual Analysis and Knowledge Representation.1993.

[3] Chandrasekaran B, Josephson JR and Benjamins VR. What are ontologies and why do we need them? IEEE Intelligent Systems. 1999; 14(1):20-26

[4] Baader F, Calvanese D, McGuinness D, Nardi D and Patel-Schneider P. The Description Logic Handbook. Cambridge University Press. 2003.

[5] BORST, P., BENJAMIN, J. WIELINGA, B and AKKERMANS, H. (1996) "An Application of Ontology Construction", ECAI-96 Workshop on Ontological Engineering, Budapest, August 13th.

[6] FERNANDEZ, M., GOMEZ-PEREZ, A. and JURISTO, N. (1997) "METHONTOLOGY: From Ontological Art Towards Ontological Engineering", AAAI-97 Spring Symposium on Ontological Engineering, Stanford University, March 24-26th.

[7] USCHOLD, M. and KING, M. (1995) "Towards A Methodology for Building Ontologies", IJCAI-95 Workshop on Basic Ontological Issues in Knowledge Sharing, Montreal, Canada.

[8] USCHOLD, M. and GR NINGER, M. (1996) "Ontologies: Principles, Methods and Applications", Knowledge Engineering Review, **11**(2), 93-137.

[9] VAN DER VET, P.E. and MARS, N.J.I. (1995) "Bottom-up Construction of Ontologies: The Case of an Ontology of Pure Substances", Technical Report 95-35, Department of Computer Science, University of Twente, Netherlands.

[10] GOMEZ-PEREZ, A., FERNANDEZ, M. and DE VICENTE, A.J. (1996) "Towards a Method to Conceptualize Domain Ontologies", ECAI-96 Workshop on Ontological Engineering, Budapest.

[11] GOMEZ-PEREZ, A., (1997). "A Framework to Verify Knowledge Sharing Technology", Expert Systems with Applications, **11**(4), 519-529.

[12] GR NINGER, M. and FOX, M.S. (1994a) "The Design and Evaluation of Ontologies for Enterprise Engineering", Workshop on Implemented Ontologies, European Conference on Artificial Intelligence 1994,, Amsterdam, NL.

[13] N. Fridman Noy, R. W. Fergerson, and M. A. Musen. The knowledge model of Prot´eg´e-2000: combining interoperability and flexibility. SMI technical report, Stanford University, School of Medicine, 2000. Submitted for publication.

[14] M. R. Genesereth and R. E. Fikes. Knowledge interchange format version 3.0 reference manual. Report Logic 92-1, Logic Group, Stanford University, California, 1992.

[15] W. E. Grosso, H. Eriksson, R. W. Fergerson, J. H. Gennari, S. W. Tu, and M. A. Musen.Knowledge modeling at the millennium: The design and evolution of Prot´eg´e-2000. In 12th Banff Workshop on Knowledge Acquisition, Modeling, and Management. Banff, Alberta, 1999.

[16] D. L. McGuinness and F. v. Harmelen, "OWL web ontology language overview: W3C recommendation 10 February 2004", OWL Web Ontology Language, W3C, 2004.

[17] J. Gennari, M. Musen, R. Fergerson, W. Grosso, M. Crub´ezy, H. Eriksson, N. Noy and S. Tu, "The evolution of Protégé 2000: An environment for knowledge-based systems development", International Journal of Human Computer Studies, 2003, 58(1):89–123.

[18] OWL Web Ontology Language http://www.w3.org/TR/owl-features/

[19] http://protege.stanford.edu