Creating an ontology using protégé: concepts and taxonomies in brief

Payam Porkar Rezaeiye[1], Mojtaba Fazli[2], Manaf Sharifzadeh[3], Hani Moghaddam[4], Mehdi Gheisari[5]

[1,2,4,5] Department of Computer, Damavand branch, Islamic Azad University, Damavand, Iran

payamporkar@damavandiau.ac.ir, mojtabafazli@yahoo.com, h.moghaddam@yahoo.com, mehdi.gheisari61@gmail.com

[3] Department of Computer, Firoozkooh branch, Islamic Azad University, Firoozkooh, Iran

manafsharif@gmail.com

Abstract:

Ontology is a discipline that has a potential improvement to information categorizing, management of information, and interpretations. The goal of ontology is representing knowledge in a way that software agents can interpret and understand data. In this paper we review important concepts of ontology creations and their related taxonomies for creation of good ontologies using protégé software.

Keywords: protégé, ontology, taxonomy, create ontology

1.Introduction:

At first sight, the goals pursued in ontology research appear to be largely the same as in classical research on knowledge representation and reasoning. An ontology defines a common vocabulary for researchers who need to share information in a domain. It includes machine-interpretable definitions of basic concepts in the domain and relations among them.

Why would someone want to develop an ontology? Some of the reasons are:

- To share common understanding of the structure of information among people or software agents
- To enable reuse of domain knowledge
- To make domain assumptions explicit
- To separate domain knowledge from the operational knowledge
- To analyze domain knowledge

Sharing common understanding of the structure of information among people or software agents is one of the more common goals in developing ontologies (Musen 1992; Gruber 1993). For example, suppose several different Web sites contain medical information or provide medical e-commerce services. If these Web sites share and publish the same underlying ontology of the terms they all use, then computer agents can extract and aggregate information from these different sites. The agents can use this aggregated information to answer user queries or as input data to other applications.

Enabling reuse of domain knowledge was one of the driving forces behind recent surge in ontology research. For example, models for many different domains need to represent the notion of time. This representation includes the notions of time intervals, points in time, relative measures of time, and so on. If one group of researchers develops such an ontology in detail, others can simply reuse it for their domains. Additionally, if we need to build a large ontology, we can integrate several existing ontologies describing portions of the large domain. We can also reuse a general ontology, such as the UNSPSC ontology, and extend it to describe our domain of interest.

Making explicit domain assumptions underlying an implementation makes it possible to change these assumptions easily if our knowledge about the domain changes. Hard-coding assumptions about the world in programming-language code make these assumptions not only hard to find and understand but also hard to change, in particular for someone without programming expertise. In addition, explicit specifications of domain knowledge are useful for new users who must learn what terms in the domain mean.

Separating the domain knowledge from the operational knowledge is another common use of ontologies. We can describe a task of configuring a product from its components according to a required specification and implement a program that does this configuration independent of the products and components themselves. We can then develop an ontology of PC-components and characteristics and apply the algorithm to configure made-to-order PCs. We can also use the same algorithm to configure elevators if we "feed" an elevator component ontology to it.

Analyzing domain knowledge is possible once a declarative specification of the terms is available. Formal analysis of terms is extremely valuable when both attempting to reuse existing ontologies and extending.

Often an ontology of the domain is not a goal in itself. Developing an ontology is akin to defining a set of data and their structure for other programs to use. Problem-solving methods, domain-independent applications, and software agents use ontologies and knowledge bases built from ontologies as data. For example, in this paper we develop an ontology of wine and food and appropriate combinations of wine with meals. This ontology can then be used as a basis for some applications in a suite of restaurant-managing tools: One application could create wine suggestions for the menu of the day or answer queries of waiters and customers. Another application could analyze an inventory list of a wine cellar and suggest which wine categories to expand and which particular wines to purchase for upcoming menus or cookbooks.

In other words:

Three simple definitions are given below.

(1) Ontology is a term in philosophy and its meaning is "Theory of existence".

(2) A definition of an ontology in AI community is "An explicit representation of Conceptualization".

(3) A definition of an ontology in KB community is "a theory of vocabulary/concepts used as building artificial systems".

Although these are compact, it is not sufficient for in-depth understanding of what an ontology is. A more comprehensive definition is given in the next subsection. In practical terms, developing an ontology includes:

- defining classes in the ontology,
- arranging the classes in a taxonomic (subclass–superclass) hierarchy,
- defining slots and describing allowed values for these slots,
- filling in the values for slots for instances.

We can then create a knowledge base by defining individual instances of these classes filling in specific slot value information and additional slot restrictions.

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.

1) *There is no one correct way to model a domain— there are always viable alternatives. The best solution almost always depends on the application that you have in mind and the extensions that you anticipate.*
2) *Ontology development is necessarily an iterative process.*
3) *Concepts in the ontology should be close to objects (physical or logical) and relationships in your domain of interest. These are most likely to be nouns (objects) or verbs*

ISBN: 978-1-61804-117-3

(relationships) in sentences that describe your domain.

2. Determine the domain and scope of the ontology

We suggest starting the development of an ontology by defining its domain and scope. That is, answer several basic question:

- What is the domain that the ontology will cover?

The answers to these questions may change during the ontology-design process, but at any given time they help limit the scope of the model.

3. Consider reusing existing ontologies

It is almost always worth considering what someone else has done and checking if we can refine and extend existing sources for our particular domain and task. Reusing existing ontologies may be a requirement if our system needs to interact with other applications that have already committed to particular ontologies or controlled vocabularies. Many ontologies are already available in electronic form and can be imported into an ontology-development environment that you are using.

4. Define the properties of classes—slots

We have already selected classes from the list of terms we created in **Error! Reference source not found.**. Most of the remaining terms are likely to be properties of these classes.

5. Define the facets of the slots

Slots can have different facets describing the value type, allowed values, the number of the values (cardinality), and other features of the values the slot can take.

6. Create instances

The last step is creating individual instances of classes in the hierarchy. Defining an individual instance of a class requires (1) choosing a class, (2) creating an individual instance of that class, and (3) filling in the slot values.

One use of formal semantics is to allow humans to reason about the knowledge. For ontological knowledge we may reason about:

_ Class membership: If x is an instance of a class C, and C is a subclass of D, then we can infer that x is an instance of D.

_ Equivalence of classes: If class A is equivalent to class B, and class B equivalent to class C, then A is equivalent to C, too.

_ Consistency: Suppose we have declared x to be an instance of the class A.

Further suppose that

- A is a subclass of B \ C
- A is a subclass of D
- B and D are disjoint

Then we have an inconsistency because A should be empty, but has the instance x. This is an indication of an error in the ontology.

_ Classification: If we have declared that certain property-value pairs are sufficient condition for membership of a class A, then if an individual x satisfies such conditions, we can conclude that x must be an instance of A.

Semantics is a prerequisite for reasoning support: Derivations such as the above can be made mechanically, instead of being made by hand.

6. Conclusion and future works:

We described how we can create ontologies and related taxonomies. We hope ontology engineering contributes to promotion of semantic web research, and hence to coping with real world problem solving.

References:

[1]C-S Lee, Z-W Jian, and L-K Huang, "A Fuzzy Ontology and Its Application to News Summarization," *(SCI) IEEE Transactions on Systems, Man and Cybernetics Part B*, vol. 35, no. 5, pp. 859-880, Oct. 2005.

[2] C-S Lee, S-M Guo, and C-Y Hsu, "Genetic-based Fuzzy Image Filter and Its Application to Image Processing," *(SCI) IEEE Transactions on Systems, Man and Cybernetics Part B*, vol. 35, no. 4, pp. 694-711, Aug. 2005.

[3] C-S Lee, C-C Jiang and Tung-Cheng Hsieh, "A Genetic Fuzzy Agent Using Ontology Model for Meeting Scheduling System," *(SCI) Information Sciences*, 2005. (Accepted)
[4] C-S Lee, Y-H Kuo, C-H Liao, and Z-W Jian, "A Chinese Term Clustering Mechanism for Generating Semantic Concepts of a News Ontology," *Journal of Computational Linguistics and Chinese Language Processing*, vol. 10, no. 2, pp. 277-302, June 2005.

[5] S-M Guo, C-S Lee, and C-Y Hsu, "An Intelligent Image Agent based on Soft-Computing Techniques for Color Image Processing," *(SCI) Expert Systems with Applications*, vol. 28, no. 3, pp. 483-494, 2005.

[6] Y-H Kuo, C-S Lee, S-M Guo, and Y-H Chen, "Apply Object-Oriented Technology to Construct Chinese News Ontology on Internet," *(EI) Journal of Internet Technology*, vol.6, no. 4, pp. 385-394, 2005.

[7] M.Gheisari," *Design, Implementation, and Evaluation of SemHD: A New Semantic Hierarchical Sensor Data Storage*", Indian journal of science and technology, March 2012