

Developing Protégé Plug-in: OWL Ontology Visualization using Social Network

Minsoo Kim* and Minkoo Kim*

Abstract: In recent years, numerous studies have been attempted to exploit ontology in the area of ubiquitous computing. Especially, some kinds of ontologies written in OWL are proposed for major issues in ubiquitous computing such like context-awareness. OWL is recommended by W3C as a descriptive language for representing ontology with rich vocabularies. However, developers struggle to design ontology using OWL, because of the complex syntax of OWL. The research for OWL visualization aims to overcome this problem, but most of the existing approaches unfortunately do not provide efficient interface to visualize OWL ontology. Moreover, as the size of ontology grows bigger, each class and relation are difficult to represent on the editing window due to the small size limitation of screen. In this paper, we present OWL visualization scheme that supports class information in detail. This scheme is based on concept of social network, and we implement OWL visualization plug-in on Protégé that is the most famous ontology editor.

Keywords: OWL visualization, Protégé, Protégé plug-in

1. Introduction

Over the years, great attention has been shown to the question of ubiquitous computing system, and some excellent results are obtained on various research areas such like sensor network, intelligent middleware and so on. For the growth of ubiquitous services and technologies, nowadays, researchers introduce and are exploiting some kinds of intelligent methods in their area. One of them, ontology has attracted a fair amount of critical attention. Especially, some famous context ontologies such like SOUPA [1] and CONON [2] are proposed to help for ensuring the ability of context-awareness for ubiquitous computing systems. Moreover domain specific ontology is widely used in the most ubiquitous research areas.

Ontology what is a formal specification of phenomenon in the world is re-focused as a logical basis for modern research issues (e.g. semantic web and smart knowledge base). Concepts, relations, instances and some restrictions are elements which compose of ontology, and concept hierarchy is the heart of it. There are some difficulties for constructing ontology caused by complex semantic of the world. To give a helping hand to ontology engineers, the guide to create ontology are even served with developing steps and domain examples [3].

Ontology language which has been researching to aim

for representing ontology easily and efficiently is one of the major issues to help for developing ontology [4]. W3C (World Wide Web Consortium) has been leading the research of ontology language, and had been proposed XML (eXtensible Markup Language) and RDF (Resource Description Framework) / RDFS (RDF Schema) as a primitive ontology language. OWL (Web Ontology Language) [5] which inherits syntax of XML and a semantic of RDF(S) are recommended by W3C as a descriptive language for ontology, and it provides rich expressive power with various useful vocabularies.

OWL makes a great contribution to the ontology engineering, but it has a difficult and complex syntax with many tags for machine understanding; this characteristic is an obstacle in developing ontology using OWL. There are various researches to solve this problem. As a result of such kind of researches, the ontology visualization is to represent ontology visually using methods such as graph. It helps us for understanding and developing OWL ontology intuitively and easily. ezOWL [6] and OWLViz [7] are well-known tools for OWL ontology visualization. They were developed as the plug-in of protégé. They try to visualize all elements in ontology using a smart graph or an enhanced tree. They give some advantages to ontology engineers, but they have critical problem due to whole ontology visualization strategy; each element in ontology is represented in a small size that users cannot recognize, since small size limitation of editor screen. This situation become serious as the size of ontology grows.

In this paper, we contribute to the OWL ontology

Manuscript received May 24, 2008; revised June 11, 2008; accepted June 26, 2008.

Corresponding Author: Minsoo Kim

* Graduate School of Information and Communication, Ajou University, Suwon, Korea ({visual, minkoo}@ajou.ac.kr)

visualization regarding partial ontology visualization. To do this, we catch a core concept of social network, and visualize classes and relations in detail. For implementing our visualization scheme, we develop visualization Protégé plug-in using Protégé APIs [8] which the famous ontology editor. We expect that our work helps to develop OWL ontology, and consequently, it makes to develop intelligent ubiquitous computing systems more easily and efficiently.

The remainders of the paper are structured as follows. Section 2 argues protégé and existing OWL visualization approaches. Section 3 describes proposed visualization scheme in detail and section 4 explains our protégé plug-in. At last the conclusions and future works are made in section 5.

2. Related Work

Before going on the explaining of our OWL visualization scheme, it might be useful to briefly consider some related works. As mentioned before, we developed the Protégé plug-in by analyzing some established visualization approaches and raising a question of whole ontology visualization in them. Therefore, we describe Protégé ontology editor and existing OWL ontology visualization approaches in this chapter.

2.1 Protégé

Protégé is an ontology editor and knowledge-based framework implemented by java for system developers and domain experts. Protégé was firstly started from a database which stores concepts, attributes and objects separately at Stanford Medical Informatics, and changed to tool for developing ontology. Currently, it is the most famous ontology editor and used by many researchers and developers. Version 3.3.1 is presented officially in August 24th, 2007.

Protégé provides familiar interfaces and many different kinds of plug-ins to users who are strange to use logical languages and logic for developing ontology. Protégé is a free and open-source platform (providing Protégé APIs). Users can develop a new plug-in easily using Protégé APIs, and the plug-in can be evaluated by someone else. This is the most worthwhile point of advantage. Protégé provides many plug-ins for ontology engineering; transformation for different languages, merging and visualization of ontology, and other useful things. Protégé users can apply any plug-ins to their application, and they can also give feedback to the plug-in developer. Especially, OWL plug-in has been developed to support characteristics having Web Ontology Language OWL such like reasoning mechanism base on

Description Logic. OWL is widely used in various research area, OWL plug-in is also used many researchers and engineers.

2.2 Ontology Visualization

Ontology can be developed by describing plain-text or using ontology editor such like protégé. However, recognition and understanding of structure of ontology is not easy with usual ways. A number of researchers make efforts to tide over this problem.

One of the challenges, ontology visualization obtained considerable results. It helps us to understand structures of ontology that has concepts, hierarchy of concepts and relationship between concepts. For the functions of visualization, there are many methods. Now, we try to explain primary methods and two famous OWL ontology visualization tools; ezOWL and OWLViz.

The ezOWL was developed by ETRI in Korea. This tool is OWL visualization plug-in for protégé. Recently, this is tried to be converting for stand-alone application. The visualization scheme of ezOWL is based on semantic networks. This network represents concepts as nodes, attributes and another feature of concepts as expanded nodes. The relationships between concepts are expressed as arcs and the name of expression is upon the arcs. The complex definitions of class or restrictions between classes are expressed by arcs and using different colors. Additionally, the tool use simple marks to express it. As we recognize the Fig. 1, the ezOWL that is visualizing OWL ontology shows ontology structure and each element on whole screen. ezOWL provides a easy way to analyze structure of ontology and understand concepts, relations and restrictions in OWL ontology.

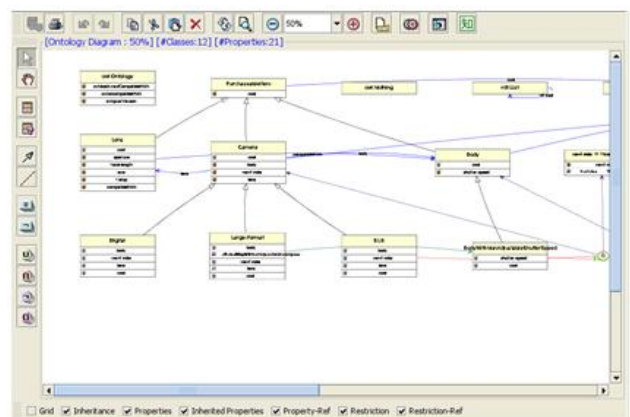


Fig. 1. Example of ezOWL visualization. It focuses on easy representation of elements in OWL

The OWLViz was developed as part of a CD-ODE project that centers on the Manchester University in UK. It

was also developed with Protégé’s plug-in and tool for the visualization of OWL ontology. At the main idea of the visualization of OWL ontology was to resolve multiple inheritance problem that was expressed in the notion of OWL ontology. The OWLViz is natively providing the graph patterns of tree to express hierarchy of concepts. The Fig. 2 shows that concepts are expressed as nodes and restricted concepts are expressed by colored nodes. The arcs are to express is-a relationships and it is able to have multiple inheritance. The features of concept and relationship are not expressed in the graphs. If we choose nodes or arcs, the tool show the detailed information with another interfaces.

Both visualization schemes try to visualize whole ontology; ontology structure, detail information of class, properties between concept, and other all information. As mentioned above, this strategy gives worthful advantages, but also has critical problem. Since a computer screen has limited size, each element in ontology should be in a small size to visualize whole ontology. In this case, users have difficulty in indicating each element. If enlarge the size of each element to avoid this problem, user should use some kind of inconvenient interface such like scroll bar. This is more serious as the size of ontology grows bigger. To solve this problem, we propose a new visualization scheme what will be discussed in next chapter.

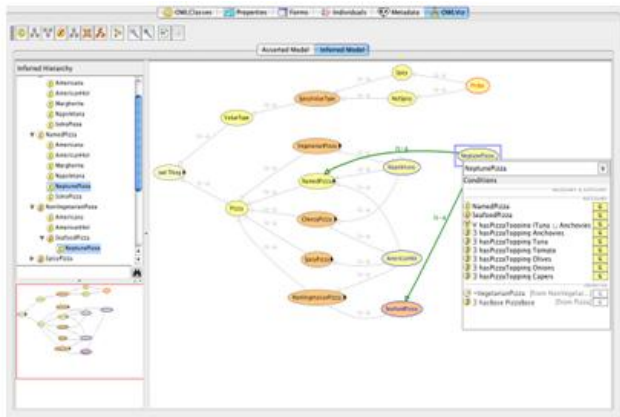


Fig. 2. Example of OWLViz visualization. It focuses on multiple inheritances of classes in OWL ontology.

3. OWL Ontology Visualization using Social Network

The visualization of whole ontology tries to visualize the structure of ontology and it is meaningful. Existing approaches for ontology visualization in section 2 target whole OWL ontology. But this approach brings about a problem that each element in ontology is represented too small and user cannot recognize it. Fig. 1 and 2 show this

problem in ezOWL and OWLViz. To get over this problem, we consider partial visualization.

3.1 Partial Ontology Visualization

Proposed method visualizes properties and neighboring classes focusing a (selected) class rather than whole ontology. It is arguable that this approach cannot show whole structure of ontology, but other component such like class tree or browser in the ontology editor can cover this problem. For example, ‘is-a’ hierarchy is clearly and simply represented in a tree structure. Users can intuitively understand the relations between visualized classes, since proposed approach visualizes the relations between specific classes.

Basically, proposed visualization (graph) represents class as a node and property as an edge. Anonymous class (i.e. composed named classes or restricted by some constraints) is represented using objects that compose the class and symbols that meet the meaning. Properties have directions from domain class to range class; object property is connected from domain class to range class, and datatype property from domain class to XML Schema Data-type (XSD) as a range class. The data-type in XSD is represented as a different-colored node distinguished from other classes. The graph is basically composed in following order.

- $c = \{ \text{selected class} \}$
- $c_i = \{ \text{one of classes represented graph (not } c) \}$
- $G = \{ \text{graph} \}$
- i) Centralizing c in G
- ii) Representing classes that associated with properties of currently selected class (c is domain and c_i is range)
- iii) Representing incoming relation from other classes to selected class (c is range and c_i is domain)
- iv) Representing *equivalentClass*, *disjointWith*, *oneOf*, *inter-sectionOf*, *unionOf*, *complementOf* on c .

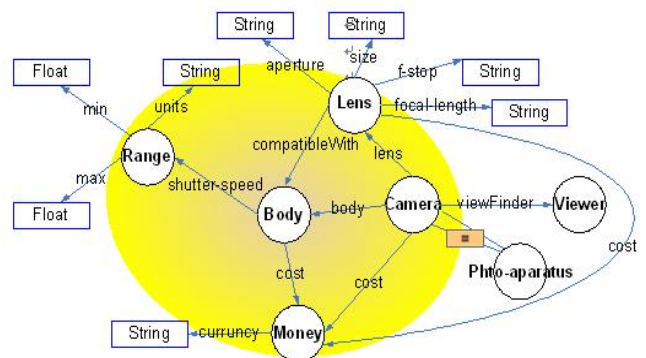


Fig. 3. Basic Graph for camera ontology (yellow-grayed circle)

Fig. 3 shows the graph of a partial camera ontology using proposed visualization scheme. Each class is represented as a white circle and white rectangle is one of the XSD. Axioms and restrictions supported by OWL are displayed using meaningful symbols. When selecting ‘Body’ class in Fig. 3, our basic graph includes elements in the grayed (yellow) circle.

3.2 Extended Graph based on Social Network

We extend basic graph to support rich representation for selected class. When the graph is extended, the weight of neighboring classes is computed, and then the classes that have weight smaller than threshold are extended. This idea is motivated the social network [10]. Social network is a social structure made of nodes which are generally individuals or organizations. It indicates the ways in which they are connected through various social familiarities ranging from casual acquaintance to close familial bonds. Applying this definition to our visualization scheme, we can say that a class can be defined by connected classes, properties, and restrictions. And a class which has a small number of properties and restrictions is strongly dependent on classes which connected by those properties and restrictions. In Fig. 3, for example, the ‘Body’ class is more important to explain ‘Range’ class rather than ‘Camera’ class, because ‘Range’ class has only one object property (only to ‘Body’) but ‘Camera’ class has many object properties and restriction. (1 arc of 5 arcs points to ‘Body’)

Exploiting this idea, we make simple weight scheme for graph extension.

$$\begin{aligned}
 o &= \{ \# \text{ object properties on } c_i : c_i \text{ is domain} \} \\
 r &= \{ \# \text{ properties from other class to } c_i : c_i \text{ is range} \} \\
 q &= \{ \# \text{ equivalentClass, disjointWith, oneOf,} \\
 &\quad \text{intersectionOf, unionOf, complementOf on } c_i \} \\
 g &= \{ \# \text{ classes in } o \text{ or } i \text{ or } q \text{ represented } G \} \\
 \text{weight of } c_i &= o + r + q - g
 \end{aligned}$$

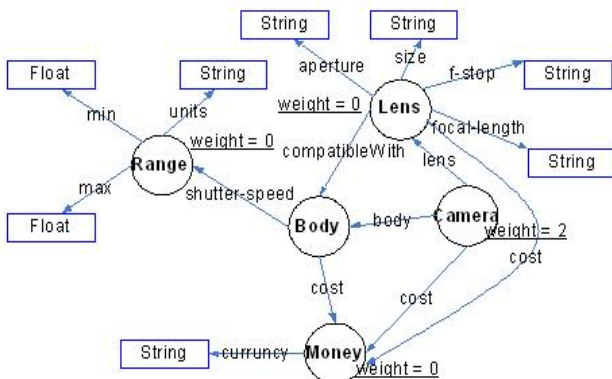


Fig. 4. Extended Graph for Figure 3 (threshold value is 1)

Datatype Property is omitted when computing the weight, because it is rather a characteristic than a relation. Fig. 4 shows a weight of classes in Fig. 3 of threshold value 1. Classes are extended in graph except ‘Camera’ class which has the weight value that does not satisfy the threshold restriction.

4. Development Protégé Plug-in

To evaluate proposed visualization scheme, we implement Protégé plug-in as a visualization topic. Protégé provides APIs to exploit their framework and to develop plug-ins. There are some advantages for developing plug-in; (1) reducing developing cost, (2) reusing well-implemented sources, (3) as the most major issue, evaluating by researchers and real users. Protégé is a open source software, plug-ins is also open and freely used and evaluated.

Fig. 5 is a screenshot for our visualization plug-in. Four main components consist of our plug-in. In the left side, class tree which is just reused on origin protégé OWL plug-in shows class hierarchies (i.e. is-a relations). Property browser is also reused, and positioned in center-top. Class tree and property browser shows all classes and properties. Graph component which is positioned in center-bottom bases on our visualization scheme. When user selects a class in a class tree, graph for the class is displayed. Right-bottom component is a description for selected class.

This is composed of two parts; description written in English and OWL source. ‘A class has property B’ is an example for English description. Description component also aims at easy understanding for ontology.

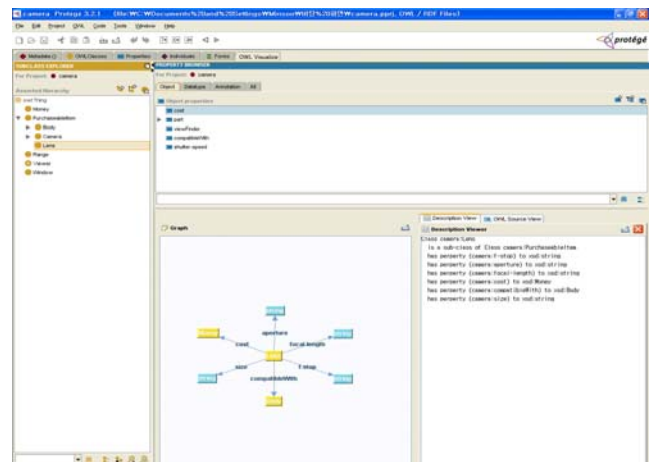


Fig. 5. Protégé Plug-in: OWL visualization based on Social Network

5. Conclusion and Future Works

This paper describes a new OWL visualization scheme based on social network. Existing OWL ontology visualization approaches which try to visualize whole ontology at a time has a critical problem due to limited screen size. To tide over the problem, we attempt to visualize OWL ontology partially. To do this, we display a detail graph for the class which is selected by user. Furthermore, by exploiting the core concept of social network, we compute weight of elements related to selected class and expand graph with more rich representation. As an implementation for proposed scheme, we develop Protégé visualization plug-in. Protégé plug-in provides several advantages; especially we can get evaluations from real users who use the plug-in. Our protégé plug-in that visualizes OWL ontology based on social network can help OWL users to provide easy way for designing and understanding ontology.

However, there are several issues that need to be considered in future works. We have to improve our weighting scheme and graph with a smart interface. To do this, we will apply well-known graph learning algorithms to our visualization scheme. And we will also reflect user (i.e. Protégé users) feedback to OWL visualization Protégé Plug-in.

Acknowledgments.

This research is supported by Foundation of ubiquitous computing and network Project (UCN), the Ministry of Knowledge Economy (MKE) 21st Century Frontier R&D Program in Korea and a result of subproject UCN 08B3-S2-10M.

Reference

- [1] Harry Chen, Tim Finin, and Anupam Joshi, "The SOUPA Ontology for Pervasive Computing", *Ontologies for Agents: Theory and Experiences*, Springer, July 2005.
- [2] Xiao Hang Wang, Tao Gu, Da Qing Zhang, Hung Keng Pung, "Ontology Based Context Modeling and Reasoning using OWL", In *Proceedings of Workshop on Context Modeling and Reasoning(CoMoRea 2004)*, In conjunction with the Second IEEE International Conference on Pervasive Computing and Communications (PerCom 2004), Orlando, Florida USA, March 2004.
- [3] Noy, N. F., McGuinness, D L.: *Ontology Development 101: a guide to creating your first ontology*. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05, Stanford Medical Informatics Technical Report SMI-2001-0880, March 2001.
- [4] Corcho, O., Gomez-Perez, A.: *A Roadmap to Ontology Specification Lan-guages*, in Rose Dieng and Olivier Corby (eds.), *Knowledge Engineering and Knowledge Management. Methods, Models and Tools*, Springer, Berlin, pages 80–96, 2000.
- [5] Web Ontology Language (OWL), <http://www.w3.org/2004/OWL/>
- [6] ezOWL Homepage, <http://iweb.etri.re.kr/ezowl/>
- [7] OWLViz User Guide, <http://www.code.org/downloads/owlviz/OWLVizGuide.pdf>
- [8] Protégé Programming Development Kit (PDK), <http://protege.stanford.edu/doc/dev.html>
- [9] Fluit, C., Sabou, M., van Harmelen, F.: *Supporting User Tasks through Visalization of Light-weight Ontologies. Hand-book on Ontologies*, Springer, Berlin, pages 425–432. 2004.
- [10] Hanneman, Robert A. and Mark Riddle. "Introduction to social network methods", Riverside, CA: University of California, Riverside, 2005.
- [11] Knublauch, H., Musen, M. A., Rector, A. L.: *Editing Description Logic Ontologies with the Protégé OWL Plugin*. In: *International Workshop on Description Logics (DL2004)*, Whistler, Canada, 2004.
- [12] Asun Gomez-Perez, Juergen Angele, Mariano Fernandez-Lopez, V. Christophides, Athur Stutt, York Sure. "A survey on ontology tools", *OntoWeb deliverable 1.3*, Universidad Politecnica de Madrid. 2002.
- [13] Protégé User Guide, <http://protege.stanford.edu/doc/users.html>
- [14] Christiaan Fluit, Marta Sabou, Frank van Harmelen. "Ontology-based Information Visualisation: Towards Semantic Web Applications", *Visualising the Semantic Web (2nd edition)*, Springer Verlag, 2005
- [15] H. Stuckenschmidt, M. Klein. "Structure-Based Partitioning of Large Concept Hierarchies", In *Proceedings of the third International Semantic Web Conference*, pp.289-303, 2004.



Minsoo Kim

He received the BS and MS degrees in Computer Science from Ajou Univ. in 2004 and 2006, respectively. He is a Ph.D. candidate as a member of the Artificial Intelligence lab at Ajou Univ., and now he is working as a visiting researcher at University of Pittsburgh,

USA. His research interests include Semantic Web, Agent Cooperation, Ontology Engineering, and Community Computing.

**Minkoo Kim**

He received a Ph.D. degree in Computer Science from PennState University, USA in 1989. From January 1999 to January 2000, he was with the University of Louisiana, CACS as a visiting researcher. Since 1981, he has been a member of the faculty in the College of Information Technology, Ajou University. His research interests are in the area of Artificial Intelligence, Multi-Agent Systems, Data Mining, Information Retrieval, Ubiquitous Computing, and Community Computing.