Class-Based OWL Ontology Visualization and Editor*

Minsoo Kim[1], JunYoung Kim[1] Seheon Song[1] Minkoo Kim[2]
[1]Graduate School of Information and Communication Ajou University, Suwon, Korea
[2]College of Information and Computer Engineering Ajou University, Suwon, Korea
[1]{bijey, tristan, legoman}@ceai.ajou.ac.kr,
[2]minkoo@ajou.ac.kr

Abstract

Although there is a growing need for editors to create ontology, most of the existing ontology editors unfortunately do not provide efficient interface to visualize ontology structure. Moreover, as the size of the ontology grows bigger, each class and relation are difficult to represent on the editing window due to the small size limitation of the editor. In this paper, we present class-based visualization scheme that supports class information in detail. we also implement efficient editor using this scheme with other components.

1 Introduction

Ontology is a logical basis for semantic web and web services. OWL is recommended by W3C as a descriptive language for web ontology. OWL is a tagging language for machine understanding, but has a difficult syntax. 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 represents ontology visually using methods such as graph[Fluit, et al., 2004]. It helps intuitive understanding of ontology and easy development of ontology.

The well-known tools for OWL ontology visualization are ezOWL and OWLViz. They were developed as the plug-in of protégé. They try to visualize whole objects in the ontology on the restricted screen. Therefore, the objects and relations are represented in a small size that users can't recognize. This situation become serious as the size of ontology grows. After all, the partial ontology visualization is necessary.

So we suggest the class-based ontology visualization method and apply this method to the OWL ontology editor.

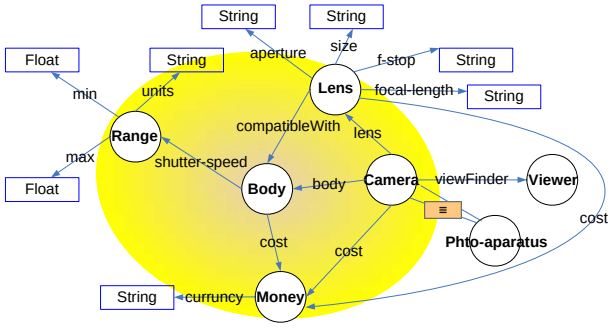
2 Class-Based Ontology Visualization

The visualization of whole ontology shows the full structure of ontology and it is meaningful. The essential part of ontology structure is 'IS-A' hierarchy of classes and properties. It is also important that recognizing the relations between classes, but the scattered graph of classes and properties on the screen is not meaningful. The visualization of whole ontology try to visualize the structure of ontology but it can't represent 'IS-A' and relations clearly.

Proposed method visualizes properties and neighboring classes focusing a class rather than whole ontology. It is arguable that this approach can't show whole structure of ontology, but other component in the ontology editor can cover this problem. For example, 'IS-A' hierarchy is clearly and simply represented in a tree structure. User can intuitively understand the relations between visualized classes, since this approach visualizes the relations between specific classes.

2.1 Basic Graph

Basically, class-based visualization represents class as a node and property as an edge. Anonymous class is represented using objects that compose the class and symbols that meet the meaning. Relations has direction from domain class to range class, the range of datatype property (XSD) is represented as a different-colored node distinguished from other relations.



[Figure 1] Graph - Camera Ontology

Graph basically composed in this order.

c = {selected class}
c_i = {one of classes represented graph (not **c**}
G = {graph}

* This research is supported by the ubiquitous Autonomic Computing and Network Project, the Ministry of Information and Communication (MIC) 21st Century Frontier R&D Program in Korea

- centralize c in G
- represent classes that associated with properties of currently selected class (c is domain and c_i is range)
- represent incoming relation from other classes to selected class (c is range and c_i is domain)
- represent *equivalentClass, disjointWith, oneOf, inter-sectionOf, unionOf, complementOf* on c.

[Figure 1] is the graph of a part of camera ontology. Yellow circle is a basic graph of 'Body' class.

2.2 Extended Graph

Graph can be extended to represent the class more in detail. When the Graph is extended, the weight of neighboring classes is computed, and then the classes that have weight smaller than threshold are extended.

o = {# object properties on c_i : c_i is domain}

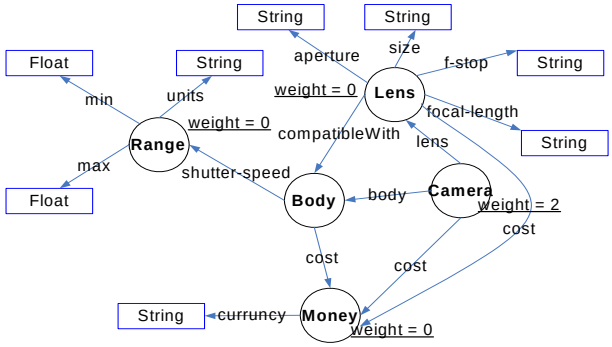
r = {# properties from other class to c_i : c_i is range}

q = {# *equivalentClass, disjointWith, oneOf, intersect tionOf, unionOf, complementOf* on c_i}

g = {# classes in o or r or q represented G}

weight of c_i = $o + r + q - g$

Datatype Property is omitted when compute the weight, because it is rather a characteristic than a relation.



[Figure 2] Extended Graph

[Figure 2] shows a weight of classes in [Figure 1] of threshold value 1. Classes are extended in graph except 'Camera' class which has the weight value that does not satisfy the threshold restriction.

3 OWL Ontology Editor

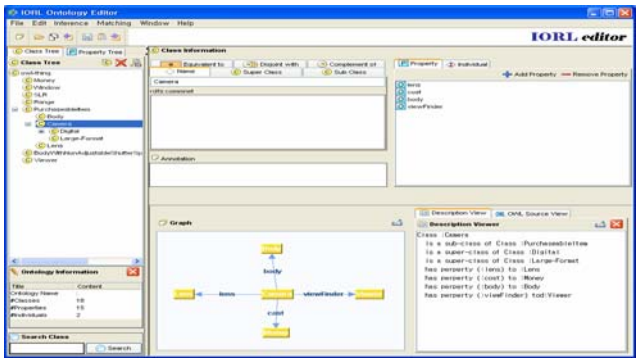
We develop an OWL ontology editor using proposed visualization method with other supporting components.

This OWL editor provides OWL ontology development environment using user friendly UI composed of Class and Property Tree, Graph, Description Viewer and also provides ontology engineering environment like working with database, automatic ontology construction. [Figure 3] is a snapshot of implemented OWL ontology editor.

Following table summarizes the main features of our ontology editor. Some are now in developing step.

Feature[1]	Contents
Modeling Features	Represent Ontology with Element based Visualization
Base Language	OWL, EOL[2)*]
Import/export	OWL, EOL[*], Database[*]
Graph View	Class-based Ontology Visualization
Consistency Checks	Not Yet
Information Extraction	Semi-Automatic Extraction for Class and property (relation) from documents[*]
Ontology Merging	Not Yet

* : developing step



[Figure 3] OWL Ontology Editor

4 Summary

Class-based ontology visualization can be a solution to the visualization of huge or complicate OWL ontology. This approach is focused on the class that user currently interested in. It represents the context of ontology and simplified structure that describes the focused class in a clear manner. As the actual application, we develop an OWL ontology editor using ontology visualization together with the Class Tree, Description Viewer. This editor supports convenient OWL ontology development environment. Moreover it provides OWL ontology engineering environment suchlike automatic ontology construction, persistent storage. This editor is an OWL ontology engineering framework also an OWL ontology editor.

References

[Fluit, et al., 2004] Christiaan Fluit, Marta Sabou and Frank van Harmelen. Supporting User Tasks through Visalization of Light-weight Ontologies. *Hand-book on Ontologies*, Springer, Berlin, pages 425--432. 2004.

[Corcho and Gomez-Perez, 2000] Oscar Corcho and Asuncion Gomez-Perez. 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.

1) [Corcho and Gomez-Perez, 2000]

2) EOL(Epistemological Ontology Language)