OWL Full Metamodeling with SWCLOS

Seiji Koide[1] and Hideaki Takeda[2]

[1] The Graduate University for Advanced Studies (SOKENDAI), 2-1-2, Hitotsubashi, Chiyoda-ku, Tokyo 101-8430 Japan
koide@nii.ac.jp,
WWW home page: http://www-kasm.nii.ac.jp/~koide/index-en.html
[2] National Institute of Informatics and SOKENDAI
takeda@nii.ac.jp

Abstract. SWCLOS is an OWL Full processor buit on top of Common Lisp Object System. We enabled OWL metamodeling with SWCLOS. In this poster and demo, we introduce criteria for metamodeling, that are derived from the principles of object-oriented metamodeling, and demonstrate examples of metamodeling with SWCLOS.

1 Introduction

Metamodeling in OWL has been discussed in the OWL 1.1 activity and other efforts [1, 2]. All of them are logic-based, and they discuss how to extend DL-based OWL to OWL Full rather than how to accomplish RDF(S) semantics in OWL. However, RDF(S) per se has potential for metamodeling. We developed SWCLOS [3], an OWL Full modeling language based on Common Lisp Object System, by leveraging RDF(S) semantics. As in RDF(S), SWCLOS provides the capability to capture a class as an individual in OWL. One still must abide by RDF(S) semantics to deal with classes as individuals.

2 Metamodeling Criteria from an OO Perspective

To capture an object as an instance, a class of an object must be established from the object-oriented (OO) perspective. This principle is extended to the class-metaclass relationship for metamodeling. Namely, in order to capture a class as an individual, we establish a class of classes (metaclass). In the object-oriented embodiment, an entity inherits the attributes and virtues of metaclasses (metaclass-hood) from a superclass as a metaclass. The source of the metaclass-hood is rdfs:Class in the RDF universe. Therefore, every metaclass must be a subclass of rdfs:Class.

Some ontologies, e.g., SUMO and Cyc, embrace embarrassing class-instance relationships, e.g., cyclic membership and disorders between classes and meta-classes. We introduce metamodeling criteria to increase the decidability while paying attention to membership classification and extension inclusiveness. If a class C is an instance of another class whose extension includes the extension of class C, we distinguish such classification from normal ones and denote the relation by $\in_{\subseteq}$ (Fig. 1). The metamodeling criteria we set up are as follows.

– If a class C is an instance of but not a subclass of D (normal), then D can be a metaclass. $CEXT^{\mathcal{I}}(D^{\mathcal{I}})$ denotes the extension of the denotation of D.

$$\{C^{\mathcal{I}} \in CEXT^{\mathcal{I}}(D^{\mathcal{I}}) \mid CEXT^{\mathcal{I}}(C^{\mathcal{I}}) \subseteq CEXT^{\mathcal{I}}(rdfs:Resource^{\mathcal{I}})\} \models CEXT^{\mathcal{I}}(D^{\mathcal{I}}) \subseteq CEXT^{\mathcal{I}}(rdfs:Class^{\mathcal{I}}) \quad (1)$$

– If a class C is an instance of and a subclass of D (abnormal), then D cannot be a metaclass.

$$\{C^{\mathcal{I}} \in_{\subseteq} CEXT^{\mathcal{I}}(D^{\mathcal{I}}) \mid CEXT^{\mathcal{I}}(C^{\mathcal{I}}) \subseteq CEXT^{\mathcal{I}}(rdfs:Resource^{\mathcal{I}})\} \not\models CEXT^{\mathcal{I}}(D^{\mathcal{I}}) \subseteq CEXT^{\mathcal{I}}(rdfs:Class^{\mathcal{I}}) \quad (2)$$

These criteria yield a guideline for metamodeling on how to resolve class-instance disorder; if a class C is a subclass of and an instance of class D (abnormal) through B that is a subclass of D, and if C is an instance of but not a subclass of class B (normal), then we can accept such an abnormal C by making B a subclass of rdfs:Class (metaclass).

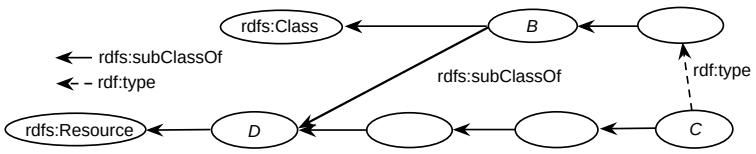

Fig. 1. Example of Meta-Modeling Criteria

3 Concluding Remarks and Demonstration

rdfs:Class plays multiple roles, as a metaclass, meta-metaclass, meta-meta-metaclass, and so forth because of its membership loop. Therefore, the above criteria create an infinite number of clearly separated layers of metamodeling, i.e., class layer, metaclass layer, meta-metaclass layer, and so forth. We demonstrate several examples of metamodeling with SWCLOS at the poster and demos.

References

1. Motik B.: On the Properties of Metamodeling in OWL. Int. Semantic Web Conf., ISWC2005 (2005) 548–562
2. Pan, J. Z., Horrocks, I.: OWL FA: a metamodeling extension of OWL DL. Proc. 15th Int. Conf. WWW, WWW '06 (2006) 1065–1066
3. Koide, S. Takeda, H.: OWL-Full Reasoning from an Object Oriented Perspective. Asian Semantic Web Conf., ASWC2006 (2006) 263–277