

UML for Ontology Development

Paul Kogut, Lockheed Martin Management & Data Systems
Stephen Cranefield, University of Otago
Lewis Hart, GRC International
Mark Dutra, Sandpiper Software, Inc.
Kenneth Baclawski and Mieczyslaw Kokar, Northeastern University
Jeffrey Smith, Mercury Computer

Introduction

Ontologies are becoming increasingly important because they provide the critical semantic foundation for many rapidly expanding technologies such as software agents, e-commerce and knowledge management (McGuinness, 2001). The Unified Modeling Language (UML)¹ has been widely adopted by the software engineering community and its scope is broadening to include more diverse modeling tasks. This paper discusses the recent convergence of UML and ontologies and suggests some possible future directions.

Current ontology research and practice had its origins in declarative AI knowledge representations such as semantic networks and frames, which were primarily designed for runtime reasoning and inference. Recent ontology efforts such as the World Wide Web Consortium (W3C) Semantic Web initiative and the DARPA Agent Markup Language (DAML)² have focused on formal logic-based and web-based knowledge representation (Berners-Lee, 2001). UML was originally designed for human-to-human communication of models for building systems in object-oriented programming languages. UML is now being used for designing artifacts that are more declarative (i.e., similar to ontologies) such as XML DTDs and Schemas (Carlson, 2001), Resource Description Framework (RDF) schemas³, database schemas (Naiburg, 2001) and knowledge models (Schreiber 1999). The Object Management Group (OMG) Model Driven Architecture (MDA)⁴ which is based on UML and related standards such as the Meta-Object Facility (MOF) and XML Metadata Interchange (XMI) is evolving to support generation of application and middleware code and translation of data in a heterogeneous environment⁵. The MDA is driving UML to become more formal and machine-processable so that models can be used at compile time and runtime and not just as a graphical notation for human-to-human communication. Researchers are developing formal semantics for UML⁶ and the MOF (Baclawski, 2001a). These efforts will remove one of the most commonly stated criticisms of the suitability of UML for representing formal models such as ontologies.

Why Use UML to Develop Ontologies?

Ontologies include class/subclass hierarchies, relationships between classes, class attribute definitions and axioms that specify constraints. In UML, this ontology information is usually modeled in class diagrams and Object Constraint Language (OCL) constraints. There are a number of good reasons why UML is a promising notation for ontologies:

- UML is an open standard maintained by the OMG. UML has standard mechanisms for defining extensions for specific application contexts such as ontology modeling.
- UML is widely adopted in industry and taught in many university courses. Current techniques for ontology development are based on knowledge representations such as Knowledge Interchange Format (KIF)⁷ which are not widely known outside the AI research community.
- UML is supported by widely-adopted CASE tools. Current ontology tools from the research community such as Ontolingua⁸ and Protege⁹, require expertise in knowledge representation.

¹ <http://www.omg.org/technology/uml/index.htm>

² <http://www.daml.org/>

³ <http://Xmodel.sourceforge.net/>

⁴ <http://www.omg.org/mda/index.htm>

⁵ <http://www.omg.org/technology/cwm/>

⁶ Precise UML Group <http://www.puml.org>

⁷ <http://logic.stanford.edu/kif/kif.html>

⁸ <http://www.ksl.stanford.edu/sns.shtml>

Recent Applications of UML for Ontology Representation

A variety of different research projects and commercial initiatives have been applying UML for ontology representation. UML can be used directly as an ontology representation or as a graphical front-end for another ontology representation language (e.g., DAML). UML has been used with a variety of implementations (e.g., Java objects, Open Knowledge Base Connectivity (OKBC)¹⁰). UML has been applied to a variety of ontology related tasks (e.g., ontology mapping, consistency checking).

Cranefield and Purvis (1999, 2000) have investigated the use of UML class diagrams for representing ontologies and UML object diagrams for representing instance knowledge. A "UML Data Binding" tool for Java (Cranefield, 2001) has been developed to generate Java classes and RDF schemas from a class diagram encoded in the XMI format. In this work, UML is used directly, not as a graphical syntax for another knowledge representation language.

Bergenti and Poggi (Bergenti, 2000) have proposed an approach to agent-oriented software engineering based on the use of UML to model various aspects of a multi-agent system¹¹. One of their proposed diagrams is an "ontology diagram", which depicts classes representing agents and domain entity types, and associations representing domain predicates that can be encoded as KIF or FIPA-SL agent message content. The UML Based Ontology Tool-set (UBOT) project¹² is building ontology engineering and natural language processing-based text annotation tools for DAML. UML is used as a front-end for visualizing and editing DAML ontologies. The approach is to extend UML by defining a UML profile for DAML which maps UML stereotypes to DAML-specific elements (Baclawski, 2001b). The UBOT tools use Telelogic Tau UML Suite for editing and generating XMI that is translated to DAML. The UBOT project has been experimenting with formal methods to check the consistency of DAML ontologies (Kokar, 2001). The UBOT tools are being evaluated in a satellite imagery analysis workflow agent application.

The Components for Ontology Driven Information Push (CODIP) project¹³ is using UML to build and map DAML ontologies in support of publish-subscribe channels for disseminating DAML messages in domains such as military logistics. Message source ontologies are mapped to message destination ontologies by creating a third ontology that is a collection of concepts and relations relating semantically equivalent concepts. The CODIP project is building a DAML-UML Enhanced Tool (DUET) which is based on Rational Rose add-ins and a prototype UML profile for DAML (Baclawski, 2001b). DUET works with an ontology articulation builder that provides automated analysis of potential mappings allowing the user to interactively build an articulation between the UML models.

Sandpiper Software¹⁴ is developing commercial grade tools that support knowledge modeling and information brokering (Dutra, 2001). Sandpiper has extended UML to enable representation of rich ontological knowledge through the creation of a UML profile for frame-based knowledge representation. They have implemented an add-in to Rational Rose that supports consistency checking and wizards that prompt the user to enter the information required. The wizards allow domain experts with little knowledge engineering expertise to build ontologies. Sandpiper has developed an OKBC-based interface for a commercial OODBMS for its knowledge base implementation. Included with Sandpiper's ontology modeling tool are base ontologies such as the IEEE Standard Upper Ontology (SUO)¹⁵.

Conclusions

UML has been successfully used for a wide range of ontology-related tasks. The OMG MDA and the W3C Semantic Web have a similar semantic interoperability vision. However, they are pursuing different metadata approaches. It is interesting to note that OMG is building a meta-metamodel (i.e., the MOF) by identifying a core set of elements from existing complex metamodels (i.e., UML and CORBA IDL) whereas the W3C is building new layers of complexity on top of simpler layers (i.e., XML -> RDF -> RDF

⁹ <http://www.smi.stanford.edu/projects/protege/>

¹⁰ <http://www.ai.sri.com/~okbc/>

¹¹ <http://www.auml.org/>

¹² <http://ubot.lockheedmartin.com/>

¹³ <http://grcinet.grci.com/maria/www/CodipSite/codip.html>

¹⁴ <http://www.sandsoft.com/>

¹⁵ <http://suo.ieee.org/>

Schema -> DAML...). There have been some efforts to bridge these two disparate metadata approaches (Melnik, 2000). However, there are serious concerns about significant semantic mismatches between the OMG and W3C approaches. Some of these significant mismatches are:

- The W3C approach does not have a clean layered architecture (Pan and Horrocks, 2001). This leads to confusing situations like a class being an instance of another class. The OMG follows the software engineering philosophy of strict separation between classes and instances.
- The OMG approach does not have a first-class concept of an "association" (analogous to a "property" in W3C terminology) (Baclawski, 2001b). Associations in UML and MOF can only exist in the context of two classes. Properties in RDF are first-class elements that can be defined in an ontology without reference to classes. If an RDF ontology states that "company owns vehicle" and "person owns dog", "owns" is the same property whereas they would be different associations in a UML model.
- The OMG logic notation, OCL, was designed to specify constraints on models with little concern for the computational complexity of runtime reasoning. The emerging W3C logic notation is being explicitly designed for tractable reasoning.

The ongoing OMG UML2.0 revision efforts and the new W3C Semantic Web Activity are good opportunities to bring the metadata approaches closer together to avoid confusion and costly workarounds.

Acknowledgments

Some of this material is based upon work supported by the Air Force Research Laboratory, Contract Number F30602-00-C-0188. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Air Force.

References

- Baclawski, Kokar, and Smith. "Metamodeling Facilities." OMG Technical Meeting, September 2001a
- Baclawski, Kokar, Kogut, Hart, Smith, Holmes, Letkowski, and Aronson. "Extending UML to Support Ontology Engineering for the Semantic Web." UML2001, October 2001b
- Bergenti, F., and A. Poggi. "Exploiting UML in the Design of Multi-Agent Systems." In *Engineering Societies in the Agents World*, Lecture Notes in Computer Science 1972 (Springer), 2000.
- Berners-Lee, Hendler, and Lassila. "The Semantic Web," *Scientific American*, May, 2001
- Carlson, D., *Modeling XML Applications with UML*. Addison-Wesley 2001
- Cranefield, S., and Purvis, M. "UML as an Ontology Modeling Language," *Proc. of the Workshop on Intelligent Information Integration, 16th Int. Joint Conference on AI (IJCAI-99)*. Germany, 1999
- Cranefield, S., and Purvis, M. "Extending Agent Messaging to Enable OO Information Exchange." in "Cybernetics and Systems 2000," R. Trappl (ed.), Vienna, Austrian Society for Cybernetic Studies (2000)
- Cranefield, S. "UML and the Semantic Web," *Proc. of the International Semantic Web Working Symposium*, Palo Alto, 2001
- Dutra, M., "UML for Knowledge Representation," OMG's Second Workshop on UML for Enterprise Applications, December 2001
- Kokar, M., "Consistency Checking of DAML+OIL." <http://vis.home.mindspring.com/>
- McGuinness, D., "Ontologies Come of Age," to appear in Fensel, Hendler, Lieberman, Wahlster editors *The Semantic Web: Why, What, and How*. MIT Press 2001
- Melnik, S. "Representing UML in RDF" <http://www-db.stanford.edu/~melnik/rdf/uml/>
- Naiburg and Maksimchuk. *UML for Database Design*. Addison-Wesley 2001
- Pan and Horrocks. "Metamodeling Architecture of Web Ontology Languages," *Semantic Web Working Symposium*, August 2001
- Schreiber, G., *Knowledge Engineering and Management*, MIT Press, 1999