Software Modeling Techniques and the Semantic Web

Jin Song Dong (dongjs@comp.nus.edu.sg)
Computer Science Department, National University of Singapore

1. Overview

Following the success of XML, W3C envisions the Semantic Web [1] as the next generation of web in which data are given well-defined and machine-understandable semantics so that they can be processed by intelligent software agents. Semantic Web can be regarded as an emerging area from the Knowledge Representation and the Web Communities. The Software Engineering community can also play an important role in the Semantic Web development. Modeling and verification techniques can be useful at many stages during the design, maintenance and deployment of Semantic Web ontology. We believe Semantic Web will be a new research and application domain for software modeling techniques and tools. For example, recent research results have shown that UML, Z [14] and Alloy [9] can provide modeling, reasoning and consistency checking services for Semantic Web.

On the other hand, the diversity of various software specification techniques and the need for their effective combinations require an extensible and integrated supporting environment. The success of the Semantic Web may have profound impact on the web environment for software design methods, especially for extending and integrating different software modeling techniques and libraries.

This tutorial will include:

- A detailed introduction to Semantic Web languages (DAML+OIL [13] and OWL [2]) and Semantic Web tools (FaCT[8] and RACER[7]).

- An introduction to software modeling techniques Z, Alloy and UML and a demonstration on how they can facilitate modeling Web ontologies.

- A combined approach to check and reasoning Web ontologies and related properties using the combination of RACER, Z/EVES [12] and the Alloy Analyser [10] tools.

- An illustration on how DAML can be used to build a Semantic Web environment for supporting, extending and integrating various software specification languages.

2. Semantic Web

The Semantic Web is a vision with a series of technologies proposed by W3C as the next generation web. It extends the current one by giving the web content a well-defined meaning, better enabling computers and people to work in cooperation [1]. Resource Description Framework (RDF) is a foundation for processing metadata; it provides interoperability between applications that exchange machine-understandable information on the Web. RDF Schema provides the basic vocabulary to describe RDF documents. RDF Schema can be used to define properties and types of the Web resources. Similar to XML Schema which give specific constraints on the structure of an XML document, RDF Schema provides information about the interpretation of the RDF statements. The DAML+OIL [13] builds on top of XML and RDF(S) to provide a language with a well-defined semantics. It has a set of language constructs, i.e., class, subclass and property with domains and ranges, for describing a Web domain. DAML can further express restrictions on membership in classes and restrictions on domains and ranges of properties. The latest Web ontology language OWL extends DAML+OIL with 3 different levels of support: OWL Lite, DL and Full, with increasing expressiveness. OWL Lite and DL are decidable whereas OWL Full is generally not. OWL Full can be regarded as an extension of RDF, while OWL Lite and OWL DL can be regarded as extensions of a restricted view of RDF. All OWL (Lite, DL, Full) documents are RDF documents. OWL has become W3C recommendation in early 2004.

As autonomous software agents may perform their reasoning without human supervision, it is essential that the shared ontology is consistent. Ontology-related reasoning tools have been developed recently. For example, FaCT [8] supports automated concept-level reasoning, namely class subsumption and consistency reasoning. RACER [7], implements a TBox (concept level) and ABox (instance level) reasoner. RACER's functionalities include creating, maintaining, deleting ontologies, concepts, roles and individuals, querying, retrieving and evaluating the knowledge base, etc.

3. Software Modeling Techniques

The software specification languages, i.e., Z [14][1] and Alloy [9][2] are suitable for specifying structural properties of software. Z and Alloy are first order declarative languages based on sets and relations. Semantic Web can be a new application domain for Z/Alloy as classes (sets of individual entities) and relationships between entities are the focused points in Semantic Web which can be directly modelled in Z and Alloy. UML's various diagrams can also be used as a design tool (class diagram) to develop Semantic Web ontology [11].

4. A Combined Reasoning Approach

Ontology reasoners such as RACER and FaCT have been developed to reason ontologies with a high degree of automation. However, complex ontology-related properties may not be expressible within the current web ontology languages, consequently they may not be checkable by RACER and FaCT. We propose to use the software engineering techniques and tools, i.e., Z/EVES [12] and Alloy Analyzer(AA) [10], to complement the ontology tools for checking Semantic Web documents [3, 6]. In this approach, Z/EVES is first applied to remove trivial syntax and type errors of the ontologies. Next, RACER is used to identify any ontological inconsistencies, whose origins can be traced by Alloy Analyzer. Finally Z/EVES is used again to express complex ontology-related properties and reveal errors beyond the modeling capabilities of the current web ontology languages. We have successfully applied this approach to checking a set of military plan ontologies.

Alternatively, if Z is used to specify the system domain and requirement models in the first place, then Web ontologies can be generated by developing a transformation program from Z to Web ontologies [5].

5. Semantic Web Linking Environment

If Semantic Web is successful, then it may become a major implementation and delivery platform for many software systems (e.g. Semantic Web Services) in the near future and it may also effect the way how software can be designed. We will demonstrate how Web ontologies can be used to build a Semantic Web environment [4] for supporting, extending and integrating various software specification languages. Furthermore, the tutorial will illustrate how RDF query techniques can facilitate specification comprehension.

References

[1] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, May 2001.

[2] M. Dean and G. Schreiber (editors). OWL Web Ontology Language Reference. W3C Recommendation, http://www.w3.org/TR/owl-ref/, 2004.

[3] J. S. Dong, C. H .Lee, Y. F. Li, and H. Wang. Verifying DAML+OIL and Beyond in Z/EVES. In *The 26th International Conference on Software Engineering (ICSE'04)*. IEEE Press, May 2004.

[4] J. S. Dong, J. Sun, and H. Wang. Semantic Web for Extending and Linking Formalisms. In L.-H. Eriksson and P. A. Lindsay, editors, *Proceedings of Formal Methods Europe: FME'02*, pages 587–606, Copenhagen, Denmark, July 2002. LNCS, Springer-Verlag.

[5] J. S. Dong, J. Sun, and H. Wang. Z Approach to Semantic Web. In C. George and H. Miao, editors, *International Conference on Formal Engineering Methods (ICFEM'02)*, pages 156–167. LNCS, Springer-Verlag, October 2002.

[6] J. S. Dong, J. Sun, and H. Wang. Checking and Reasoning about Semantic Web through Alloy. In *Proceedings of 12th Internation Symposium on Formal Methods Europe: FM'03*, pages 796–813, Pisa, Italy, September 2003. LNCS, Springer-Verlag.

[7] Volker Haarslev and Ralf Möller. *RACER User's Guide and Reference Manual: Version 1.7.6*, December 2002.

[8] I. Horrocks. The FaCT system. *Tableaux'98, Lecture Notes in Computer Science*, 1397:307–312, 1998.

[9] D. Jackson. Micromodels of software: Lightweight modelling and analysis with alloy. Available: http://sdg.lcs.mit.edu/alloy/book.pdf (an early version has been published in TOSEM Vol-11), 2002.

[10] D. Jackson, I. Schechter, and I. Shlyakhter. Alcoa: the alloy constraint analyzer. In *Proc. 22nd International Conference on Software Engineering: ICSE'2000*, pages 730–733, Limerick, Ireland, June 2000. ACM Press.

[11] P. Kogut, S. Cranefield, L. Hart, M. Dutra, K. Baclawski, M. Kokar, and J. Smith. UML for Ontology Development. *Knowledge Engineering Review*, 17, 2002.

[12] M. Saaltink. The Z/EVES system. In J. P. Bowen, M. G. Hinchey, and D. Till, editors, *ZUM'97: Z Formal Specification Notation*, volume 1212 of *Lecture Notes in Computer Science*, pages 72–85. Springer-Verlag, 1997.

[13] F. van Harmelen, P. F. Patel-Schneider, and I. Horrocks (editors). Reference description of the daml+oil ontology markup language. Contributors: T. Berners-Lee, D. Brickley, D. Connolly, M. Dean, S. Decker, P. Hayes, J. Heflin, J. Hendler, O. Lassila, D. McGuinness, L. A. Stein, ..., March, 2001.

[14] J. Woodcock and J. Davies. *Using Z: Specification, Refinement, and Proof*. Prentice-Hall International, 1996.

---

1 Z is based on set theory and predicate calculus and has been developed at Oxford University. The use of Z for specifying software systems has resulted in a number of successful stories, including the 1992 Queen's Award for its use in the IBM CICS project and the recent Praxis' successful commercial project on smart cards development.

2 Alloy is a structural modelling language based on first-order logic, for expressing complex structural constraints and behaviour.