From Semantic Web to Expressive Specifications: A Modeling languages Spectrum

Jin Song Dong
School of Computing, National University of Singapore
dongjs@comp.nus.edu.sg

Abstract

Many researchers at W3C currently focus on developing the next generation of the Web — the Semantic Web. The development of the Web ontology languages, RDF, OWL and SWRL, is reminiscent of the early development of system specification languages in software engineering communities. Indeed, from the expressiveness point of view, Web ontology languages are subsets of Alloy, UML/OCL, VDM, Z and Object-Z. One can futher predict that the modeling languages for capturing the behaviours of the Semantic Web Services and Agents can be drawn from the rich collections of software dynamic modeling techniques, i.e., state machines, process algebra and integrated design methods. This tutorial will present a concise Modeling Languages Spectrum that includes a few key representative modeling languages ranging from simple static Web Ontology modeling techniques to expressive dynamic integrated modeling techniques. Comparisons and transformations between those languages will be discussed. Furthermore, based on transformation approaches, the latest research results on applying software modeling techniques and tools to the Semantic Web domain will be also demonstrated.

D.2.4 Software/Program Verification

Categories and Subject Descriptors

D.2.1 [**Requirements/Specifications**]: Languages; D.2.4 [**Software/Program Verification**]: Formal methods

General Terms

Languages, Verification

1. GOAL AND OBJECTIVES

The purpose of this tutorial is to incrementally introduce various modeling techniques from Semantic Web domain and Software Specification domain, and also to present the state of the art of the latest research and development of this new joint area on Semantic Web and Software Engineering.

ICSE'06, May 20–28, 2006, Shanghai, China.
ACM 1-59593-085-X/06/0005.

2. BACKGROUND AND OBSERVATION

W3C (www.w3c.org) has successfully pushed XML in the main stream of software industries. Currently one of the main W3C's effort is to develop the Semantic Web [2] as the next generation of the Web in which data is given well-defined and machine-understandable semantics so that it can be processed by intelligent software agents. Data are defined in terms of ontologies, which capture concepts and relationships. Ontology languages such as RDF [17] and OWL [3] provide basic vocabularies for describing Web resources. Recently W3C has concentrated on developing Semantic Web Rule Languages (SWRL) [13] that further extends OWL with more expressive language constructs. The future research topics, i.e., Semantic Web Services and Agents, will certainly shift the W3C research focus from the static data modeling towards dynamic behaviour modeling.

The development of the Web ontology languages, RDF, OWL and SWRL, is reminiscent of the early development of specification and design languages in software engineering communities. Indeed, from the expressiveness point of view, RDF, OWL and SWRL are subsets of Alloy [14], UML/OCL [22], VDM [16], Z [24] and Object-Z [8]. One can further predict that the modeling languages for capturing the behaviours of the Semantic Web Services and Agents can be drawn from the rich collections of formal behavioural modeling techniques, i.e., state machines (e.g. statechart[10], timed automata [1]), process algebra (e.g. CSP [11], Pi-calculus [19]) and (even possibly) integrated formalisms (e.g. Circus [23], TCOZ [18]).

3. MODELING LANGUAGE SPECTRUM

Given the wide range of modeling techniques from the Semantic Web and Software Engineering communities, one main part of this tutorial will present a concise Modeling Languages Spectrum that includes a few key representative modeling languages ranging from simple static data Web ontology modeling techniques to expressive dynamic integrated modeling techniques. Comparisons between those languages will be discussed.

4. SOFTWARE REASONING TOOLS FOR SEMANTIC WEB

Semantic Web not only emerges from the Knowledge Representation and the Web Communities, but also brings the two communities closer together. 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, the current Web Ontology reasoners such as FaCT [12] and RACER [9] 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. In this tutorial, we propose to use the software engineering techniques and tools, i.e., Z/EVES [20] and Alloy Analyzer(AA) [15], to complement the ontology tools for checking Semantic Web documents [6, 7, 21]. In our 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.

5. ANALYSING MODELS THROUGH TRANSFORMATIONS

Transformation techniques between different languages are useful and important not only for checking Web ontology through software modeling languages and tools, but also for checking and analysing complex software design models. For example, integrated formal modeling techniques are well suited for presenting more complete and coherent requirement models for complex systems. However, the challenge is how to analyze and check these models with tools support. We believe one effective approach is to project (transform) the integrated requirement models into multiple domains, then to use existing specialized tools in those domains to perform the checking and analyzing tasks. This tutorial will also address some of the research issues in the transformation techniques between different modeling techniques [5, 4].

6. REFERENCES

[1] R. Alur and D. L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126:183–235, 1994.

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

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

[4] J. S. Dong, P. Hao, and B. Mahony. Formal designs for embedded and hybrid systems. *International Journal on Software Engineering and Knowledge Engineering*, 15(2):373–378, April 2005.

[5] J. S. Dong, P. Hao, S.C. Qin, J. Sun, and Y. Wang. Timed Patterns: TCOZ to Timed Automata. *the 6th International Conference on Formal Engineering Methods (ICFEM'04)*, pages 483–498. Springer, November 2004.

[6] 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)*. ACM/IEEE Press, May 2004.

[7] 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, September 2003. LNCS, Springer.

[8] R. Duke and G. Rose. *Formal Object Oriented Specification Using Object-Z*. Cornerstones of Computing Series. Macmillan, March 2000.

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

[10] D. Harel. On Visual Formalisms. *Communications of the ACM*, 31(5), 1988.

[11] C.A.R. Hoare. *Communicating Sequential Processes*. Prentice-Hall, 1985.

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

[13] I. Horrocks, P.F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean. SWRL: A Semantic Web Rule Language Combining OWL and RuleML. http://www.w3.org/Submission/2004/SUBM-SWRL-20040521/, 2004.

[14] D. Jackson. Micromodels of software: Lightweight modelling and analysis with alloy. Available: http://sdg.lcs.mit.edu/alloy/book.pdf, 2002.

[15] 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.

[16] C. Jones. *Systematic Software Development Using VDM*. Prentice-Hall, 1986.

[17] O. Lassila and R. R. Swick (editors). Resource description framework (rdf) model and syntax specification. http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/, Feb 1999.

[18] B. Mahony and J. S. Dong. Timed Communicating Object Z. *IEEE Transactions on Software Engineering*, 26(2):150–177, February 2000.

[19] R. Milner. *Communicating and mobile systems : the π-calculus*. Cambridge University Press, 1999.

[20] 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 *LNCS*, Springer, 1997.

[21] H. Wang, J. S. Dong, and J. Sun. Reasoning Support for SWRL-FOL Using Alloy. In *17th SEKE'05*, July 2005.

[22] J. Warmer and A. Kleppe. *The object constraint language: precise modeling with UML*. Addison-Wesley, Boston, MA, USA, 1999.

[23] J. Woodcock and A. Cavalcanti. The Semantics of Circus. In *2nd International Conference on Z and B*, volume 2272 of *LNCS*, Springer, 2002.

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