Chapter 8

Conclusion

This chapter serves two purposes. Firstly, a conclusion of the whole thesis is given, summarizing the main contributions and secondly, a discussion on future work directions is also presented.

8.1 Main Contributions of the Thesis

Ontology languages form the foundation of the Semantic Web and they are of utter importance to the upper-layer technologies in the Semantic Web, such as Web services, trust modeling, etc.

As the Semantic Web is envisioned as a ubiquitous network for humans as well as machines, software agents can cooperate and aggregate Web resources from different sites to carry out complex tasks autonomously. Hence, automation of core reasoning tasks performed by agents is very important. It is for this reason ontology languages such as DAML+OIL and OWL are designed to be decidable.

Decidability is achieved by limiting the expressivity of ontology languages.

Being based on description logics, a subset of first-order predicate logic, DAML+OIL and OWL statements can only express properties with a limited degree of complexity. Many desirable properties cannot be represented in these languages. Such a challenge is often faced by Semantic Web developers as it is often the case that complex properties capture vital information pertaining to the validity of the ontology are too complex to be modeled in DAML+OL or OWL, even in its most expressive species OWL Full.

The newly proposed rules extension to OWL, the SWRL, partially solves the problem by incorporating Horn-style clauses into OWL.

Being able to represent the complex properties is only the first step. The ability to reason about ontologies and associated complex properties efficiently is at least as important. However, as SWRL is undecidable, there is unlikely that a proof tool can support all reasoning tasks for SWRL ontologies.

This thesis presents my research works in answering some of these challenges. The five main contributions of this thesis can be summarized as follows.

- We identified the expressivity limitation of ontology languages and defined a Z semantics (in Chapter 3) for the ontology languages DAML+OIL and OWL, making it possible to use software engineering proof tools such as Z/EVES to perform complex reasoning tasks on Semantic Web ontologies. We have shown that properties crucial to the validity of an ontology can be checked by Z/EVES. Some of these properties are inexpressible in ontology languages, even in SWRL.
- Based on the above work, we proposed a combined approach in Chapter 4, exploiting the complementary power of software engineering proof tools such as Z/EVES and Alloy Analyzer and Semantic Web reasoning engines such as

RACER and FaCT++. The application of these tools in combination can verify the correctness of DAML+OIL/RDF ontologies and debug inconsistent ontologies more effectively.

RACER and other Semantic Web reasoning engines are fully automated. Given an ontology, these reasoners can judge whether it is consistent without user interaction. However, as stated previously, the automation is based on the fact that the expressivity of ontology languages is limited. Hence, complex properties inexpressible in these languages are certainly un-checkable by these tools. Moreover, these description logics-based tools can only detect that there is an inconsistency in the ontology, they cannot tell where and how this is caused, making debugging large ontologies very hard.

Alloy Analyzer is an automated constraint solver with the ability of finding the source of the error if there is one. This ability is achieved by giving a finite scope to each Alloy specification to be solved by Alloy Analyzer. This fits naturally with Semantic Web reasoning engines as Alloy Analyzer can be used like a surgery tool to precisely locate the source of the inconsistencies found by Semantic Web reasoning engines.

Theorem provers such as Z/EVES are very powerful and they can prove complex properties that ontology languages and Alloy cannot represent. Hence, Z language is used to represent complex properties about ontologies and Z/EVES is used to perform a final proof of such complex properties interactively.

The above combined approach has been successfully applied to a military planning ontologies case study, where one ontological inconsistency was discovered and located and a number of errors undetected by RACER and Alloy Analyzer were found by Z/EVES.

- The applicability of the above combined approach largely relies on the soundness of the Z/Alloy semantics for DAML+OIL and OWL DL. Hence, it is important to formally prove the soundness of these semantics. As OWL is based on DAML+OIL and it has been recommended as *the* ontology language, we have developed a Z semantics for OWL DL, a sub language of OWL that is most comparable with DAML+OIL.

As ontology languages and Z are based on different logical systems, a more abstract device that is able to represent and inter-relate different logical systems is needed to formally investigate their relationship. Institutions were introduced to formalize the notion of logical systems. Institution morphisms provide means of translating signatures of different institutions while preserving *truth*. Hence, institutions and institution morphisms are natural candidates to prove the soundness of Z semantics for OWL DL (hence DAML+OIL). In Chapter 5, we have defined institutions $\mathfrak{O}$ (for OWL DL) and $\mathfrak{Z}$ (for Z) and used institution comorphisms to prove the soundness of the above semantics.

- To ease the application of the combined approach, we have developed a tools environment, the SESeW. Chapter 6 presented this environment in detail.

  SESeW implements the ontology development methodology, the Methontology [29] to systematically create an ontology. Given a number of terms in a particular domain, a user can create an OWL ontology by following some simple steps to designate terms to OWL classes, properties and individuals and relate them.

  With an ontology, SESeW can perform a number of tasks. Firstly, a user can transform it into specifications in various formal languages such as Alloy, Z and PVS [79] fully automatically. Secondly, a user can query the ontology by issuing RDQL [86] queries in the friendly interface provided by SESeW. A number of query templates in the military domain have been created for non-expert users. These templates simplifies the querying process by hiding non-necessary technical details.

  Moreover, SESeW serves as a point of contact to the various external editing and reasoning tools such as OilEd, RACER, Alloy Analyzer and Z/EVES. It can also invoke functionalities of RACER directly to check the consistency of a given DAML+OIL or OWL ontology.

- In the development of the Semantic Web, a services ontology, the OWL Services (OWL-S), has been developed to add semantic information to the Web services. This is one step closer to realize the full potential of the Web.

  This thesis presented an approach to visualize and simulate Semantic Web services (OWL-S) [95] ontologies using Live Sequence Charts (LSCs) [18] and Play-Engine [38].

  The OWL-S ontology was developed to complement Web Services standards such WSDL [14] to semantically markup the capabilities, requirements, control constructs, inputs/outputs, preconditions and effects of Web services.

  As OWL-S ontologies capture dynamic aspects of Web services, the core reasoning services, namely subsumption, consistency and instantiation, are no longer adequate to ensure their correctness.

  In Chapter 7, we translate OWL-S process models to Live Sequence Charts and use Play-Engine to visualize and simulate them. By "playing out" the charts, potential undesired scenarios can be detected early, without actually implementing the services.

In summary, our research in this thesis attempts to answer some of the challenges in the realization of the Semantic Web vision by representing and proving complex ontology-related properties using a combination of software engineering and Semantic Web techniques synergistically. It also opens up a new application domain for software engineering languages and tools.

8.2 Future Work Directions

Based on the works in this thesis, there are a number of directions of future research that may be beneficial to the Semantic Web community. In this section, some of these possible research works are briefly discussed.

8.2.1 Further Development of SESeW

As presented in Chapter 6, the SESeW tools environment is developed to ease the application of the combined approach. Still in a prototype stage, there is room for improvement. Based on the feedback from users, we will further improve it in the following aspects:

Support of up-to-date RDF query engine Recently, a more sophisticated query RDF language, the SPARQL [81] has been developed to replace RDQL. How SESeW can support this query language is also a future research work.

Support of rules extension of OWL As we mentioned in the overview in Chapter 2, SWRL [48] has been accepted by the W3C as a member submission. It is layered on top of OWL to improve the expressivity of the Semantic Web languages. It is very likely for SWRL to be officially integrated into the Semantic Web. Hence, it is necessary to keep SESeW updated with the technology trend.

The Z semantics for SWRL has been developed in Chapter 4. The Alloy and PVS semantics for SWRL can be similarly defined. By incorporating transformation procedures into SESeW, SWRL ontologies can be checked using software engineering tools such as Z/EVES, Alloy Analyzer and the PVS theorem prover [78].

With the support of SWRL, we can look into SWRL FOL [9], an extension of SWRL towards full first-order logic. With SWRL FOL, being a part of the combined approach and SESeW, expressive power of Z and Alloy can be tapped by translating Z theorems and/or Alloy assertions and facts into SWRL FOL ontologies. By doing so, software engineering practitioners can work with ontologies with greater ease.

Tighter integration with external tools Some of the external reasoners used in the combined approach such as Alloy Analyzer and RACER provide Java-based APIs, which can be used to make direct function calls from within SESeW, e.g., calling reasoning functions from SESeW directly without invoking the GUI of RACER to determine the consistency a given ontology.

This improvement has already been experimented where from SESeW, we can already invoke RACER's methods to check the consistency of a given DAML+OIL/OWL ontology.

More flexible support for ontology query Currently SESeW supports ontology query with built-in query templates particularly geared towards the military plan ontologies. It is our development plan that users are able to create, modify and delete query templates in a future version.

8.2.2 Verification of Web Ontologies – Beyond Static Data

Semantically marked-up data on the Web alone cannot fulfill the full potential of the Semantic Web. These data must be machine-interpretable and machine-processable. Web Services, enable users to effect changes in the world. Built on top of OWL, the OWL Services ontology OWL-S [95] provides semantic markup for low-level service description languages. Looking into the issue on how software engineering techniques and tools can benefit SW Services is another promising future research direction.

Chapter 7 presented our research of using Live Sequence Charts and Play-Engine to model, visualize and simulate OWL-S process models. With no open XML textual representation of LSCs, the transformation from OWL-S to LSC is a manual process.

Model checking techniques [15] may prove to be applicable in this domain. Berghofer and Nipkow [75] have recently developed a tool for Isabelle/HOL [76] that supports random testing of specifications, which may be useful in specifying and verifying Web services ontologies. The Communicating Sequential Processes (CSP) [42] is a well-known event-based formal notation primarily aimed at describing the sequencing of behavior within a process and the synchronization of behavior between different processes. FDR (Failures-Divergence Refinement) [83] is a CSP model checker that verifies CSP models automatically. It also provides a graphical interface for determining the source of errors by analyzing the trace of events that led up to the error. Other model checkers such as SPIN [43] may also be used.

Symbolic Analysis Laboratory (SAL) [6] is a framework for combining different tools for abstraction, program analysis, theorem proving and model checking. towards the symbolic analysis of concurrent systems expressed as transition systems.

SAL defines a a common intermediate language to describe transition systems. This intermediate language serves as a common medium from which various analysis tools such as the PVS theorem prover and SMV [70] model checker can be invoked by translating the intermediate language to the specific language used by these tools.

We believe that SAL can be a candidate environment for reasoning Web service ontologies. Besides theorem proving and model checking, SAL specifications can also be translated to Java code for animation purposes. By developing translators to translate Web service ontologies to the SAL common intermediate language, the above reasoning services can all be readily deployed.

8.2.3 Augmenting the Semantic Web with Belief

As the Web is a constantly evolving and totally distributed environment, software agents may from time to time face incomplete, incoherent or incomplete data. This is especially the case when the agent needs to aggregate data developed or maintained by different sites. It will be valuable for agents in these situations to associate belief with Web resources.

Currently, all ontology languages in the Semantic Web stack, such as RDF Schema, OWL and SWRL, are based on crisp logics, in which all statements are interpreted to be either true or false. Hence, the lack of the ability of associating confidence factors with ontology statements is another prominent expressivity limitation of the current ontology languages.

We believe that by extending ontology languages to allow fuzzy or belief-based interpretations of statements will help to resolve the above problem. Belief Augmented Frames (BAF) [93] is an extension to the Minsky knowledge representation systems [72]. In BAF, concepts are represented by frames and relations between concepts are represented by slots. We associate a pair of values representing belief/disbelief values with each frame and slot. BAF-logic defines how the two values are calculated and combined to give the confidence factor of a certain frame/slot. The belief/disbelief values are obtained independent from each other, allowing for greater flexibility in modeling ignorance and confidence.

The other future research direction that is worth to pursuit is to integrate BAF with OWL and RDF to incorporate belief factors into the Semantic Web stack.

Bibliography

[1] J. Angele, H. Boley, J. de Bruijn, D. Fensel, P. Hitzler, M. Kifer, R. Krummenacher, H. Lausen, A. Polleres, and R. Studer. Web Rule Language (WRL), Version 1.0, 2005. http://www.wsmo.org/wsml/wrl/wrl.html.

[2] S. Battle, A. Bernstein, H. Boley, B. Grosof, M. Gruninger, R. Hull, M. Kifer, D. Martin, S. McIlraith, D. McGuinness, J. Su, and S. Tabet. Semantic Web Services Language (SWSL). http://www.daml.org/services/swsf/1.0/swsl/, 2005.

[3] H. Baumeister. Relating abstract datatypes and Z-schemata. In D. Bert and C. Choppy, editors, *Recent Trends in Algebraic Development Techniques - Selected Papers*, volume 1827 of *Lect. Notes in Comput. Sci.*, pages 366–382, Bonas, France, 2000. Springer-Verlag.

[4] S. Bechhofer, I. Horrocks, C. Goble, and R. Stevens. OilEd: a reason-able ontology editor for the semantic web. In *Proceedings of KI2001, Joint German/Austrian conference on Artificial Intelligence*, number 2174 in Lecture Notes in Computer Science, pages 396–408, Vienna, September 2001. Springer-Verlag.

[5] S. Bechhoffer. The dig description logic interface: Dig/1.1. Technical report, The University Of Manchester, The University Of Manchester, Oxford Road, Manchester M13 9PL, 2003.

[6] S. Bensalem, V. Ganesh, Y. Lakhnech, C. M. noz, S. Owre, H. Rueß, J. Rushby, V. Rusu, H. Saïdi, N. Shankar, E. Singerman, and A. Tiwari. An overview of SAL. In C. M. Holloway, editor, *LFM 2000: Fifth NASA Langley Formal Methods Workshop*, pages 187–196, Hampton, VA, jun 2000. NASA Langley Research Center.

[7] T. Berners-Lee. Uniform Resource Identifiers (URI): Generic Syntax. http://www.ietf.org/rfc/rfc2396.txt, 1998.

[8] T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, 284(5):35–43, 2001.

[9] H. Boley, M. Dean, B. Grosof, I. Horrocks, P. Patel-Schneider, S. Tabet, and G. Wagner. SWRL FOL (November 2004). http://www.daml.org/2004/11/fol/, Nov. 2004.

[10] D. Booth, M. Champion, C. Ferris, F. McCabe, E. Newcomer, and D. Orchard. Web Services Architecture. http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/, Feb. 2004.

[11] J. P. Bowen and M. G. Hinchey. Seven more myths of formal methods. *IEEE Software*, 12(4):34–41, 1995.

[12] J. P. Bowen and M. G. Hinchey. Ten Commandments of Formal Methods. *IEEE Computer*, 28(4):56–63, 1995.

[13] J. Broekstra, M. Klein, S. Decker, D. Fensel, and I. Horrocks. Adding formal semantics to the web: building on top of rdf schema. In *ECDL Workshop on the Semantic Web: Models, Architectures and Management*, 2000.

[14] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana. *Web Services Description Language (WSDL) 1.1*. W3C, 1.1 edition, March 2001. http://www.w3c.org/TR/wsdl.

[15] E. Clarke. Automatic verification of finite state concurrent systems using temporal logic specifications: A practical approach. In *Proc. 10th ACM Symp. on Princ. Prog. Lang.*, pages 117–127, 1983.

[16] E. Clarke and J. Wing. Formal methods: State of the art and future directions. *ACM Computing Surveys*, 28(4):626–643, Dec. 1996. Other working group members: R. Alur, R. Cleaveland, D. Dill, A. Emerson, S. Garland, S. German, J. Guttag, A. Hall, T. Henzinger, G. Hozmann, C. Jones, R. Kurshan, N. Leveson, K. McMillan, J. Moore, D. Peled, A. Pnueli, J. Rushbby, N. Shankar, J. Sifakis, P. Sistla, B. Steffen, P. Wolper, J. Woodcock, and P. Zave.

[17] D. Brickley and R.V. Guha (editors). Resource description framework (rdf) schema specification 1.0. http://www.w3.org/TR/rdf-schema/, Feb. 2004.

[18] W. Damm and D. Harel. LSCs: Breathing Life into Message Sequence Charts. In *Proceedings of the IFIP TC6/WG6.1 Third International Conference on Formal Methods for Open Object-Based Distributed Systems (FMOODS)*, page 451. Kluwer, B.V., 1999.

[19] J. de Bruijn, D. Fensel, P. Hitzler, M. Kifer, and A. Polleres. Relationship of WRL to relevant other technologies. http://www.w3.org/Submission/WRL-related/, Sept. 2005.

[20] J. de Bruijn, R. Lara, A. Polleres, and D. Fensel. Owl dl vs. owl flight: conceptual modeling and reasoning for the semantic web. In A. Ellis and T. Hagino, editors, *WWW*, pages 623–632. ACM, 2005.

[21] J. S. Dong, Y. Feng, and Y. F. Li. Reasoning Support for the OWL Rules Language. In *Proceedings of First International Colloquium on Theoretical Aspects of Computing (ICTAC'04)*, Guiyang, China, Sept. 2004.

[22] J. S. Dong, Y. Feng, Y. F. Li, and J. Sun. A tools environment for developing and reasoning about ontologies. In *Proc. of 12th Asia-Pacific Software*

Engineering Conference (APSEC'05), Taipei, Taiwan, Dec. 2005.

[23] J. S. Dong, C. H. Lee, Y. F. Li, and H. Wang. A Combined Approach to Checking Web Ontologies. In *Proceedings of 13th World Wide Web Conference (WWW'04)*, pages 714–722, New York, USA, May 2004.

[24] J. S. Dong, C. H. Lee, Y. F. Li, and H. Wang. Verifying DAML+OIL and beyond in Z/EVES. In *Proceedings of 26th International Conference on Software Engineering (ICSE'04)*, pages 201–210, Edinburgh, Scotland, May 2004.

[25] J. S. Dong, Y. F. Li, J. Sun, J. Sun, and H. Wang. XML-based static type checking and dynamic visualization for TCOZ. In *International Conference on Formal Engineering Methods (ICFEM'02)*, pages 311–322, Shanghai, China, Oct. 2002. LNCS, Springer-Verlag.

[26] J. S. Dong, Y. F. Li, and H. Wang. TCOZ Approach to Semantic Web Services Design. In *Proceedings of 13th World Wide Web Conference (WWW'04)*, pages 442–443, New York, USA, May 2004.

[27] J. S. Dong, J. Sun, and H. Wang. Checking and Reasoning about Semantic Web through Alloy. In *Proceedings of Formal Methods Europe: FME'03*, volume 2805 of *Lect. Notes in Comput. Sci.*, pages 796–814, Pisa, Italy, Sept. 2003. LNCS, Springer-Verlag.

[28] R. Duke, G. Rose, and G. Smith. Object-Z: a Specification Language Advocated for the Description of Standards. *Computer Standards and Interfaces*, 17:511–533, 1995.

[29] M. Fernandez, A. Gomez-Perez, and N. Juristo. METHONTOLOGY: from Ontological Art towards Ontological Engineering. In *Proceedings of the AAAI97 Spring Symposium Series on Ontological Engineering*, pages 33–40, Stanford, USA, Mar. 1997.

[30] J. Gennari, M. A. Musen, R. W. Fergerson, W. E. Grosso, M. Crubézy, H. Eriksson, N. F. Noy, and S. W. Tu. The evolution of protégé: An environment for knowledge-based systems development. Technical Report SMI-2002-0943, Stanford Medical Informatics, Stanford University, 2002.

[31] J. Goguen and R. M. Burstall. Introducing institutions. In *Proc. Logics of Programming Workshop*, number 164 in Lect. Notes in Comput. Sci., pages 221–256. Springer-Verlag, 1984.

[32] J. Goguen and R. M. Burstall. Institutions: Abstract Model Theory for Specification and Programming. *Journal of the Association for Computing Machinery*, 39(1):95–146, Jan. 1992. Predecessor in: LNCS 164, 221–256, 1984.

[33] J. Goguen and G. Roşu. Institution morphisms. Formal Aspects of Computing, 2002.

[34] T. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.

[35] V. Haarslev and R. Möller. Practical Reasoning in Racer with a Concrete Domain for Linear Inequations. In I. Horrocks and S. Tessaris, editors, *Proceedings of the International Workshop on Description Logics (DL-2002)*, Toulouse, France, Apr. 2002. CEUR-WS.

[36] V. Haarslev and R. Möller. *RACER User's Guide and Reference Manual: Version 1.7.6*, Dec. 2002.

[37] D. Harel and R. Marelly. Specifying and Executing Behavioral Requirements: The Play-In/Play-Out Approach. Technical Report MCS01-15, The Weizmann Institute of Science Rehovot, Israel, 2002.

[38] D. Harel and R. Marelly. *Come, Let's Play: Scenario-Based Programming Using LSCs and the Play-Engine*. Springer-Verlag, 2003.

[39] I. Hayes, editor. *Specification Case Studies*. International Series in Computer Science. Prentice-Hall, 1987.

[40] J. Heflin. OWL Web Ontology Language Use Cases and Requirements. http://www.w3.org/TR/2004/REC-webont-req-20040210/, Feb. 2004.

[41] M. Hinchey and J. P. Bowen, editors. *Applications of Formal Methods*. Prentice Hall, 1995.

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

[43] G. Holzmann. The model checker spin. *IEEE Trans. on Software Engineering*, 23(5):279–295, May 1997. Special issue on Formal Methods in Software Practice.

[44] I. Horrocks. Fact++ web site. http://owl.man.ac.uk/factplusplus/.

[45] I. Horrocks. The FaCT system. *Tableaux'98, LNCS*, 1397:307–312, 1998.

[46] I. Horrocks. DAML+OIL: a description logic for the semantic web. *IEEE Data Engineering Bulletin*, 25(1):4–9, 2002.

[47] I. Horrocks and P. F. Patel-Schneider. A proposal for an owl rules language. In *Proc. of the Thirteenth International World Wide Web Conference (WWW 2004)*, pages 723–731, New York, USA, May 2004. ACM. http://www.cs.man.ac.uk/~horrocks/DAML/Rules/.

[48] 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/, May 2004.

[49] I. Horrocks, P. F. Patel-Schneider, and F. van Harmelen. From $\mathcal{SHIQ}$ and RDF to OWL: The making of a web ontology language. *J. of Web Semantics*, 1(1):7–26, 2003.

[50] I. Horrocks, U. Sattler, and S. Tobies. Practical reasoning for very expressive description logics. *Logic Journal of the IGPL*, 8(3):239–263, 2000.

[51] HP Labs. Jena Semantic Web Toolkit - version 1. http://www.hpl.hp.com/semweb/jena1.htm.

[52] ISO/IEC 13568:2002. Information technology—Z formal specification notation—syntax, type system and semantics, 2002. International Standard.

[53] ITU. *Message Sequence Chart(MSC)*, Nov 1999. Series Z: Languages and general software aspects for telecommunication systems.

[54] 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.

[55] D. Jackson, I. Schechter, and I. Shlyakhter. Alcoa: the Alloy Constraint Analyzer. In *The 22nd International Conference on Software Engineering (ICSE'00)*, pages 730–733, Limerick, Ireland, June 2000. ACM Press.

[56] Jos de Bruijn and Axel Polleres and Rubén Lara and Dieter Fensel. OWL$^-$. http://www.wsmo.org/wsml/wrl/wrl.html, 2004. Deliverable D20.1v0.2, WSML.

[57] H. Knublauch, R. W. Fergerson, N. F. Noy, and M. A. Musen. The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications. In *Proceedings of the Third International Semantic Web Conference (ISWC 2004),*, Hiroshima, Japan, Nov. 2004.

[58] P. Lambrix. Description Logics home page. http://www.ida.liu.se/labs/iislab/people/patla/DL/index.html.

[59] S. M. Lane. *Categories for the Working Mathematician*. Springer-Verlag, New York, second edition, 1985.

[60] C. H. Lee. Phase I Report for Plan Ontology. DSO National Labs, Singapore, 2002.

[61] Y. F. Li, J. Sun, G. Dobbie, J. Sun, and H. Wang. Validating Semistructured Data using OWL. In *Proceedings of the 7th International Conference on Web-Age Information Management (WAIM'06)*, Hong Kong, China, June 2006.

[62] S. Liu. *A Structured and Formal Requirements Analysis Method Based on Data Flow Analysis and Rapid Prototyping*. PhD thesis, The University of Manchester, 1992.

[63] D. Lucanu, Y. F. Li, and J. S. Dong. Soundness proof of Z semantics of OWL using institutions. In *Fourteenth International Conference on World Wide Web (WWW'05)*, pages 1048–1049, Chiba, Japan, May 2004.

[64] D. Lucanu, Y. F. Li, and J. S. Dong. Institution Morphisms for Relating OWL and Z. In *Proc. of The 17th International Conference on Software Engineering and Knowledge Engineering (SEKE'05)*, Taipei, Taiwan, July 2005.

[65] D. Lucanu, Y. F. Li, and J. S. Dong. Semantic web languages – towards an institutional perspective. In K. F. et al., editor, *Algebra, Meaning and Computation, Festschrift in Honor of Prof. Joseph Goguen*, volume 4060 of *Lect. Notes in Comput. Sci.*, pages 99–123. Springer-Verlag, 2006. to appear.

[66] M. Dean and G. Schreiber (editors). OWL Web Ontology Language Reference. http://www.w3.org/TR/2004/REC-owl-ref-20040210/, Feb. 2004.

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

[68] F. Manola and E. M. (editors). RDF Primer. http://www.w3.org/TR/rdf-primer/, Feb. 2004.

[69] D. L. McGuinness and F. van Harmelen (editors). OWL Web Ontology Language Overview. http://www.w3.org/TR/2003/PR-owl-features-20031215/, Dec. 2003.

[70] K. L. McMillan. *Symbolic Model Checking*. Kluwer Academic Publishers, 1993.

[71] I. Meisels and M. Saaltink. The Z/EVES Reference Manual (for Version 1.5). Technical Report TR-97-5493-03d, ORA Canada, One Nicholas Street, Suite 1208 - Ottawa, Ontario K1N 7B7 - CANADA, Sept. 1997.

[72] M. Minsky. A framework for representing knowledge. In J. Haugeland, editor, *Mind Design: Philosophy, Psychology, Artificial Intelligence*, pages 95–128. MIT Press, Cambridge, MA, 1981.

[73] M. Moser, O. Ibens, R. Letz, J. Steinbach, C. Goller, J. Schumann, and K. Mayr. SETHEO and E-SETHEO – the CADE-13 systems. *Journal of Automated Reasoning*, 18:237–246, 1997.

[74] D. Nardi and R. J. Brachman. An introduction to description logics. In F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, editors, *The description logic handbook: theory, implementation, and applications*, pages 1–40. Cambridge University Press, 2003.

[75] S. B. T. Nipkow. Random testing in Isabelle/HOL. In *Proceedings of the 2nd Software Engineering and Formal Methods (SEFM 2004)*, Beijing, China, Sept. 2004. IEEE Computer Society Press. To appear.

[76] T. Nipkow, L. C. Paulson, and M. Wenzel. *Isabelle/HOL — A Proof Assistant for Higher-Order Logic*, volume 2283 of *LNCS*. Springer, 2002.

[77] Ontoweb Ontology-Based Information. Deliverable 1.3: A survey on ontology tools. http://ontoweb.aifb.uni-karlsruhe.de/About/Deliverables/D13_v1-0.zip.

[78] S. Owre, J. M. Rushby, and N. Shankar. PVS: A prototype verification system. In D. Kapur, editor, *11th International Conference on Automated Deduction (CADE)*, volume 607 of *Lecture Notes in Artificial Intelligence*, pages 748–752, Saratoga, NY, June 1992. Springer-Verlag.

[79] S. Owre, N. Shankar, J. M. Rushby, and D. W. J. Stringer-Calvert. *PVS Language Reference*. Computer Science Laboratory, SRI International, Menlo Park, CA, Dec. 2001.

[80] P. F. Patel-Schneider and I. Horrocks (editors). OWL: Direct Model-Theoretic Semantics. http://www.w3.org/TR/owl-semantics/direct.html.

[81] E. Prud'hommeaux and A. Seaborne. SPARQL Query Language for RDF. http://www.w3.org/TR/2006/CR-rdf-sparql-query-20060406/, Apr. 2006.

[82] A. Riazanov and A. Voronkov. The design and implementation of vampire. *AI Communications*, 15(2-3):91–110, 2002.

[83] A. W. Roscoe. *Theory and Practice of Concurrency*. International Series in Computer Science. Prentice-Hall, 1997.

[84] 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 *Lect. Notes in Comput. Sci.*, pages 72–85. Springer-Verlag, 1997.

[85] M. Saaltink. The Z/EVES 2.0 User's Guide. Technical Report TR-99-5493-06a, ORA Canada, One Nicholas Street, Suite 1208 - Ottawa, Ontario K1N 7B7 - CANADA, Oct. 1999.

[86] A. Seaborne. RDQL - A Query Language for RDF, Jan. 2004. http://www.w3.org/Submission/2004/SUBM-RDQL-20040109/.

[87] M. Sintek and S. Decker. TRIPLE—A query, inference, and transformation language for the semantic web. In I. Horrocks and J. Hendler, editors, *The Semantic Web — ISWC 2002. Proceedings of the First International Semantic Web Conference*, volume 2348 of *Lect. Notes in Comput. Sci.*, pages 364–378, Sardinia, Italy, June 2002. Springer-Verlag.

[88] J. M. Spivey. *Understanding Z: A Specification Language and its Formal Semantics*, volume 3 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 1988.

[89] J. M. Spivey. *The Z Notation: A Reference Manual*. International Series in Computer Science. Prentice-Hall, 2nd edition, 1992.

[90] J. Sun, Y. F. Li, H. Wang, and J. Sun. 'Visualizing and Simulating Semantic Web Services Ontologies. In *Proc. of 7th International Conference on Formal Engineering Methods (ICFEM'05)*, pages 439–445, Manchester, UK, Nov. 2005. LNCS, Springer-Verlag.

[91] J. Sun, H. Zhang, Y. F. Li, and H. Wang. Formal Semantics and Verification for Feature Modeling. In *Proc. of 10th IEEE International Conference on Engineering of Complex Computer Systems (ICECCS'05)*. IEEE Press, June 2005.

[92] T. Tammet. Gandalf. *Journal of Automated Reasoning*, 18(2):199–204, Apr. 1997.

[93] C. K. Y. Tan. *Belief Augmented Frames*. PhD thesis, National University of Singapore, 2003.

[94] A. Tarlecki. Moving between logical systems. In M. Haveraaen, O. Owe, and O.-J. Dahl, editors, *Recent Trends in Data Type Specifications. 11th Workshop on Specification of Abstract Data Types*, volume 1130 of *Lect. Notes in Comput. Sci.*, pages 478–502. Springer-Verlag, 1996.

[95] The OWL Services Coalition. OWL-S: Semantic Markup for Web Services. http://www.daml.org/services/owl-s/, 2004.

[96] Tim Berners-Lee. cwm - a general purpose data processor for the semantic web. http://www.w3.org/2000/10/swap/doc/cwm, 2004.

[97] D. Tsarkov and I. Horrocks. DL reasoner vs. first-order prover. In *Proc. of the 2003 Description Logic Workshop (DL 2003)*, volume 81 of *CEUR (http://ceur-ws.org/)*, pages 152–159, 2003.

[98] D. Tsarkov and I. Horrocks. Efficient reasoning with range and domain constraints. In *Proc. of the 2004 Description Logic Workshop (DL 2004)*, pages 41–50, 2004.

[99] UDDI. *Universal Description, Discovery, and Integration of Business for the Web*, October 2001. http://www.uddi.org.

[100] F. van Harmelen and D. Fensel. Formal methods in knowledge engineering. *The Knowledge Engineering Review*, 10(4):345–360, 1995.

[101] F. van Harmelen, P. F. Patel-Schneider, and I. H. (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, et. al., March, 2001.

[102] H. Wang. *Semantic Web and Formal Design Methods*. PhD thesis, National University of Sinagpore, 2004.

[103] H. Wang, J. S. Dong, J. Sun, and Y. F. Li. TCOZ Approach to OWL-S Process Model Design. In *Proc. of The 17th International Conference on Software Engineering and Knowledge Engineering (SEKE'05)*, Taipei, Taiwan, July 2005.

[104] H. Wang, Y. F. Li, J. Sun, and H. Zhang. Verify Feature Models using Protégé-OWL. In *Fourteenth International Conference on World Wide Web (WWW'05)*, pages 1038–1039, Chiba, Japan, May 2004.

[105] H. Wang, Y. F. Li, J. Sun, H. Zhang, and J. Pan. A Semantic Web Approach to Feature Modeling and Verification. In *1st Workshop on Semantic Web Enabled Software Engineering (SWESE'05)*, Galway, Ireland, Nov 2005. LNCS, Springer-Verlag. accepted.

[106] J. Woodcock and S. Brien. $\mathcal{W}$: A Logic for Z. In *Proceedings of Sixth Annual Z-User Meeting*, University of York, Dec 1991.

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

[108] World Wide Web Consortium (W3C). Extensible Markup Language (XML). http://www.w3.org/XML.

[109] World Wide Web Consortium (W3C). XSL Transformations (XSLT) Version 1.0. http://www.w3.org/TR/xslt, 1999.

[110] World Wide Web Consortium (W3C). *Simple Object Access Protocol (SOAP) 1.1*, 2000. http://www.w3c.org/TR/SOAP.