The Distributed Ontology, Model and Specification Language – DOL

Till Mossakowski

The Distributed Ontology, Model and Specification Language – DOL

Till Mossakowski

Institute of Intelligent Cooperating Systems,
Otto-von-Guericke-University Magdeburg, Germany

Over the last decades, the WADT community has studied the formal specification of software (and hardware) in great detail [9, 1, 42]. One important aspect is the structuring of specifications in a modular way [43], which has been covered in specification languages like CLEAR [6], OBJ [18], ASL [46] and many others. Here, a powerful abstraction is the notion of institution, introduced by Goguen and Burstall [17]. It enables the study of concepts and languages for structured specifications in a way that is completely independent of the underlying logical system — the only condition being that the logical system is formalised as an institution, which is a rather mild requirement. Such an institution independent kernel language for structured specifications has been introduced in [41], and based on this, later the Common algebraic specification language CASL [3, 37] has been standardised.

While all these developments, including CASL, focus on formal *specifications*, the approach of providing an institution-independent language for the structuring of logical theories (or more precisely, finite presentations of these) can be applied to other areas as well.

In particular, in research on *ontologies*, the notion of conservative extension has been cited from the algebraic specification literature (e.g. [25]) und used for the notion of ontology module extraction in various description logics (see e.g. [21], and [19] for an institution-independent generalisation). The existing multitude of ontology languages like OWL and its sublogics, RDF, RDFS and their relations have been captured using institutions [23, 32].

Moreover, using the notion of heterogeneous multi-logic specification developed in [2, 12, 44, 14, 27, 28, 22, 36], a program for the institution-based formalisation of UML multi-viewpoint *models* has been formulated [8, 20, 7]. Note that *model* here is to be understood in the sense of model-driven engineering (MDE), to be distinguished from models in the sense of logical model theory (and institutional specification theory). In order to avoid confusion, we henceforth call the former *MDE models*.

Based on this observation of similarities between ontologies, MDE models and specifications, the Distributed Ontology, Model and Specification Language (DOL) has been proposed and adopted as an OMG standard [38, 33, 31]. **O**ntologies, MDE **m**odels and **s**pecifications are commonly abbreviated by the acronym OMS. Hence, DOL can be seen as a language for building OMS in a structured way and expressing their relations. CASL already provides several structuring constructs, e.g. (possibly conservative or definitional) extensions, unions, translations and hidings. DOL extends these in several ways:

theory-level semantics CASL uses a model-theoretic semantics, that is, a specification denotes a signature and a class of models over that signature. DOL adopts this, but also features theory-level semantics [40, 42] for certain constructs like module extraction or filtering.

reduction CASL features hiding of a specification (aka OMS) along a signature morphism, corresponding to the restriction to an export interface. DOL features three more similar operations:

- **module extraction** extraction of a sub-OMS such that the original OMS is a conservative extension [21]. The extracted module may extend the given restriction signature.
- **approximation** gives the theorems visible over the restriction signature and corresponds to the theory-level semantics of hiding [40, 42]. The problem of capturing this theory by a finite presentation has been studied for ontology languages under the terms *forgetting* and *uniform interpolation* [45, 24].
- **filtering** extraction of a sub-OMS consisting of all sentences that actually are formed over the restricted signature [39].

minimization whereas free specifications in CASL allow the selection of the least intepretation of e.g. predicates, minimization allows the selection of all minimal interpretations, following McCarthy's circumscription [26]. Also, the duals (cofree and maximal OMS) are included. Cofree OMS can be used for coinductive specification of process types, like in CoCASL [34].

refinement simple refinements are specification morphisms [42] (logically: interpretations of theories [15], in terms of OBJ [18] and CASL [3, 37]: views). The refinement language of [30] is included into DOL, that is, certain operation on refinements are available, like composition and extension. However, neither architectural specifications nor branching refinements are included, because their semantics is still subject of ongoing research ([11] had not been available when the DOL standard emerged).

equivalence OMS can be declared to equivalent, if they have a common definitional extension [35, 22]

alignment this notion is a relational generalisation of signature morphisms (which are typically functional in nature) [16, 13, 47]. Between a symbol from the source OMS and one from the target OMS, different relations can be specified.

networks networks generalise distributed specifications [35], networks of alignments [16] and distributed description logics [4]. They provide also a formal notion of viewpoint specifications, e.g. collections of UML diagrams providing different views on a system. A model of a network is a family of models of the involved OMS that is compatible along the mappings of the network. Networks can also be refined.

combination When alignments are normalised to spans or Ws of signature morphisms, networks correspond to diagrams (in the sense of category theory) of OMS [10]. A network can be combined into a single OMS by taking its colimit. Under suitable amalgamation conditions, the combination captures the model class of the network and thus can be used for reasoning about networks.

entailments between OMS, or of an OMS by a network.

heterogeneity support for multiple logics (institutions) as discussed above: OMS can be translated along institution comorphisms, be projected along institution morphisms. Also, approximations, refinements and alignments can be heterogeneous.

internet compatibility all names are full URLs resp. IRIs, and prefix maps allow the convenient abbreviation of these.

This completes the overview of DOL, which is currently being finalised. The DOL standard document is available at `omg.org/spec/DOL`; further information can be found at `dol-omg.org`. Tool support for (an increasing part of) DOL is provided by the Heterogeneous Tool Set (`hets.eu`) and Ontohub (`ontohub.org`). Sample DOL documents can be found at `ontohub.org/dol-examples`.

Future work will address the further extension of DOL, e.g. with queries and architectural refinements. Also, the extension of proof support from standard structured specifications [5, 29] to the whole of DOL is an important task.

Acknowledgements The author wishes to thank the community that has developed DOL, in particular Mihai Codescu, Michael Gruninger, Maria Keet, Alexander Knapp, Oliver Kutz, Christoph Lange and Fabian Neuhaus, as well as those OMG members that have provided valuable feedback, in particular Conrad Bock, Elisa Kendall, Pete Rivett and Ed Seidewitz.

References

1. E. Astesiano, H.-J. Kreowski, and B. Krieg-Brückner. *Algebraic Foundations of Systems Specification*. Springer, 1999.
2. G. Bernot, S. Coudert, and P. Le Gall. Towards heterogeneous formal specifications. In *AMAST 96*, volume 1101 of *LNCS*, pages 458–472. 1996.
3. M. Bidoit and Peter D. Mosses. *CASL — the Common Algebraic Specification Language: User Manual*. Number 2900 in LNCS. Springer Verlag, 2004.
4. A. Borgida and L. Serafini. Distributed Description Logics: Assimilating Information from Peer Sources. *Journal of Data Semantics*, 1:153–184, 2003.
5. Tomasz Borzyszkowski. Logical systems for structured specifications. *Theoretical Computer Science*, 286:197–245, 2002.
6. Rod M. Burstall and Joseph A. Goguen. The semantics of CLEAR, a specification language. In Dines Bjørner, editor, *Abstract Software Specifications 1979, Proceedings*, volume 86 of *LNCS*, pages 292–332. Springer, 1979.
7. Daniel Calegari, Till Mossakowski, and Nora Szasz. Heterogeneous verification in the context of model driven engineering. *Sci. Comput. Program.*, 126:3–30, 2016.
8. M. V. Cengarle, A. Knapp, A. Tarlecki, and M. Wirsing. A heterogeneous approach to UML semantics. In P. Degano, R. De Nicola, and J. Meseguer, editors, *Concurrency, Graphs and Models, Essays Dedicated to Ugo Montanari on the Occasion of His 65th Birthday*, LNCS 5065, pages 383–402. Springer, 2008.

9. M. Cerioli, M. Gogolla, H. Kirchner, B. Krieg-Brückner, Z. Qian, and M. Wolf. *Algebraic System Specification and Development. A Survey and Annotated Bibliography, 2nd Edition*, volume 3 of *BISS monographs*. Shaker Verlag, 1997.
10. Mihai Codescu, Till Mossakowski, and Oliver Kutz. A categorical approach to ontology alignment. Submitted.
11. Mihai Codescu, Till Mossakowski, Donald Sannella, and Andrzej Tarlecki. Specification refinements: calculi, tools, and applications. *Science of Computer Programming*, in press.
12. S. Coudert, G. Bernot, and P. Le Gall. Hierarchical heterogeneous specifications. In J. L. Fiadeiro, editor, *WADT 1998*, LNCS 1589, pages 106–120. Springer, 1999.
13. Jérôme David, Jérôme Euzenat, François Scharffe, and Cássia Trojahn dos Santos. The alignment API 4.0. *Semantic Web*, 2(1):3–10, 2011.
14. Razvan Diaconescu. Grothendieck Institutions. *Applied Cat. Struct.*, 10:383–402, 2002.
15. H. B. Enderton. *A Mathematical Introduction to Logic*. Academic Press, 1972.
16. J. Euzenat and P. Shvaiko. *Ontology matching*. Springer, 2nd edition, 2013.
17. J. A. Goguen and R. M. Burstall. Institutions: Abstract model theory for specification and programming. *Journal of the Association for Computing Machinery*, 39:95–146, 1992. Predecessor in: LNCS 164, 221–256, 1984.
18. J. A. Goguen, C. Kirchner, H. Kirchner, A. Mégrelis, J. Meseguer, and T. C. Winkler. An introduction to OBJ 3. In Stéphane Kaplan and Jean-Pierre Jouannaud, editors, *Conditional Term Rewriting Systems*, LNCS 308, pages 258–263, 1987.
19. Y. A. Ibáñez-García, T. Mossakowski, D. Sannella, and A. Tarlecki. Modularity of ontologies in an arbitrary institution. In N. Martí-Oliet, P. Csaba Ölveczky, and C. L. Talcott, editors, *Logic, Rewriting, and Concurrency - Essays dedicated to José Meseguer on the Occasion of His 65th Birthday*, LNCS 9200, pages 361–379, 2015.
20. A. Knapp, T. Mossakowski, and M. Roggenbach. Towards an institutional framework for heterogeneous formal development in UML — A position paper. In R. De Nicola and R. Hennicker, editors, *Software, Services, and Systems — Essays Dedicated to Martin Wirsing on the Occasion of His Retirement from the Chair of Programming and Software Engineering*, LNCS 8950, pages 215–230, 2015.
21. Boris Konev, Carsten Lutz, Dirk Walther, and Frank Wolter. Formal properties of modularisation. In Heiner Stuckenschmidt, Christine Parent, and Stefano Spaccapietra, editors, *Modular Ontologies: Concepts, Theories and Techniques for Knowledge Modularization*, volume 5445 of *LNCS*, pages 25–66. Springer, 2009.
22. Oliver Kutz, Till Mossakowski, and Dominik Lücke. Carnap, Goguen, and the Hyperontologies: Logical Pluralism and Heterogeneous Structuring in Ontology Design. *Logica Universalis*, 4(2):255–333, 2010.
23. D. Lucanu, Y.-F. Li, and J. S. Dong. Semantic Web Languages – Towards an Institutional Perspective. In K. Futatsugi, J.-P. Jouannaud, and J. Meseguer, editors, *Algebra, Meaning, and Computation, Essays Dedicated to Joseph A. Goguen on the Occasion of His 65th Birthday*, LNCS 4060, pages 99–123, 2006.
24. Carsten Lutz and Frank Wolter. Foundations for Uniform Interpolation and Forgetting in Expressive Description Logics. In Toby Walsh, editor, *IJCAI*, pages 989–995. IJCAI/AAAI, 2011.
25. T. S. E. Maibaum. Conservative extensions, interpretations between theories and all that! In Michel Bidoit and Max Dauchet, editors, *TAPSOFT'97: Theory and Practice of Software Development*, LNCS 1214, pages 40–66. Springer, 1997.
26. John McCarthy. Circumscription - a form of non-monotonic reasoning. *Artif. Intell.*, 13(1–2):27–39, 1980.

27. T. Mossakowski. Comorphism-based Grothendieck logics. In K. Diks and W. Rytter, editors, *MFCS 2002*, volume 2420 of *LNCS*, pages 593–604. Springer, 2002.
28. T. Mossakowski. Heterogeneous specification and the heterogeneous tool set. Habilitation thesis, University of Bremen, 2005.
29. T. Mossakowski, S. Autexier, and D. Hutter. Development graphs – proof management for structured specifications. *Journal of Logic and Algebraic Programming*, 67(1-2):114–145, 2006.
30. T. Mossakowski, D. Sannella, and A.Tarlecki. A simple refinement language for CASL. In J. L. Fiadeiro, editor, *WADT 2004*, LNCS 3423, pages 162–185. 2005.
31. Till Mossakowski, Mihai Codescu, Fabian Neuhaus, and Oliver Kutz. The distributed ontology, modelling and specification language - DOL. In A. Koslow and A. Buchsbaum, editors, *The Road to Universal Logic–Festschrift for 50th birthday of Jean-Yves Beziau, Volume II*, Studies in Universal Logic. Birkhäuser, 2015.
32. Till Mossakowski and Oliver Kutz. The Onto-Logical Translation Graph. In Oliver Kutz and Thomas Schneider, editors, *Modular Ontologies*. IOS, 2011.
33. Till Mossakowski, Oliver Kutz, Mihai Codescu, and Christoph Lange. The Distributed Ontology, Modeling and Specification Language. In C. Del Vescovo, T. Hahmann, D. Pearce, and D. Walther, editors, *Proceedings of the 7th International Workshop on Modular Ontologies (WoMO-13)*, CEUR-WS 1081, 2013.
34. Till Mossakowski, Lutz Schröder, Markus Roggenbach, and Horst Reichel. Algebraic-co-algebraic specification in COCASL. *Journal of Logic and Algebraic Programming*, 67(1-2):146–197, 2006.
35. Till Mossakowski and Andrzej Tarlecki. Heterogeneous logical environments for distributed specifications. In Andrea Corradini and Ugo Montanari, editors, *WADT 2008*, volume 5486 of *LNCS*, pages 266–289. Springer, 2008.
36. Till Mossakowski and Andrzej Tarlecki. A relatively complete calculus for structured heterogeneous specifications. In Anca Muscholl, editor, *FOSSACS 2014*, volume 8412 of *LNCS*, pages 441–456. Springer, 2014.
37. Peter D. Mosses, editor. *CASL Reference Manual*. Number 2960 in LNCS. Springer Verlag, 2004.
38. Object Management Group. The distributed ontology, modeling, and specification language (DOL), 2016. OMG standard available at `http://www.omg.org/spec/DOL`.
39. Florian Rabe and Michael Kohlhase. A scalable module system. *Information & Computation*, 230(1):1–54, 2013.
40. Grigore Roşu and Joseph Goguen. Composing hidden information modules over inclusive institutions. In Olaf Owe, Stein Krogdahl, and Tom Lyche, editors, *From Object-Orientation to Formal Methods. Essays in Memory of Ole-Johan Dahl*, volume 2635 of *LNCS*, pages 96–123. Springer, 2004.
41. D. Sannella and A. Tarlecki. Specifications in an arbitrary institution. *Information and Computation*, 76:165–210, 1988.
42. Donald Sannella and Andrzej Tarlecki. *Foundations of Algebraic Specification and Formal Software Development*. EATCS Monographs in Theoretical Computer Science. Springer, 2012.
43. Donald Sannella and Martin Wirsing. Specification languages. In *Algebraic Foundations of Systems Specification* [1], pages 243–272.
44. A. Tarlecki. Towards heterogeneous specifications. In D. Gabbay and M. de Rijke, editors, *Frontiers of Combining Systems 2, 1998*, Studies in Logic and Computation, pages 337–360. Research Studies Press, 2000.
45. Zhe Wang, Kewen Wang, Rodney W. Topor, and Jeff Z. Pan. Forgetting for knowledge bases in DL-Lite. *Ann. Math. Artif. Intell.*, 58(1–2):117–151, 2010.

46. Martin Wirsing. Structured algebraic specifications: A kernel language. *Theor. Comput. Sci.*, 42:123–249, 1986.
47. Antoine Zimmermann, Markus Krötzsch, J. Euzenat, and Pascal Hitzler. Formalizing Ontology Alignment and its Operations with Category Theory. In *Proc. of FOIS-06*, pages 277–288, 2006.