The Heterogeneous Tool Set — some recent developments and highlights

Till Mossakowski[1] and Mihai Codescu[2]

[1] Institute of Intelligent Cooperating Systems,
Otto-von-Guericke-University Magdeburg, Germany
[2] Institute of Mathematics "Simion Stoilow" of the Romanian Academy,
Research Group of the project PED-0494, Bucharest, Romania

The is a large body of literature on modular structured and architectural specification, refinement, colimits, proof calculi etc. [3, 13, 4, 1, 2, 12, 14], which develops these notions (at the level of definitions, theorems and also language design) independently of the underlying logic. The logic at hand needs to be formalised as an institution. The notion of institution is a powerful abstraction, introduced by Goguen and Burstall [7]. The satisfaction condition imposed by the notion of institution is a rather mild requirement met by most logical systems.

The Heterogeneous Tool Set (Hets [11]) is a parsing, analysis and proof management tool for heterogeneous specifications, available at `hets.eu`. It provides a software interface for plugging-in institutions, and provides tool support for modular specification and refinement in an institution-independent way, thus complementing the theoretical institution-independent approach in the literature with a practical tool. Moreover, based on a graph of institutions and institution translations (formalised as comorphisms [8]), Hets supports heterogeneous multi-logic specifications as developed in [15, 6, 9].

In this work, we update the ten-year old description of Hets in [11] with recent highlights and developments.

Distributed Ontology, Model and Specification Language (DOL) Originally, Hets has used the Common Algebraic Specification Language CASL [2, 12] as input language. CASL provides institution-independent structuring constructs, and can be used with any institution as base logic. Based on CASL, we have developed the Distributed Ontology, Model and Specification Language (DOL) [10], see also `omg.org/spec/DOL` and `dol-omg.org`. DOL extends CASL several ways: several notions of reduction (including ontology module extraction and uniform interpolation), minimization (following McCarthy's circumscription), alignments (a kind of relational theory morphisms), as well as networks of specifications and their colimits (the latter is called combination in DOL).

Internet and Linked Data Compatibility While in CASL, identifiers are essentially strings (and internet compatibility is only given indirectly via downloads of libraries), Hets now supports arbitrary URLs/IRIs. Specifications and specification libraries can directly accessed via their URL.

Architectural Refinement Simple refinements are specification morphisms [14] (logically: interpretations of theories). They are available in CASL [2, 12] as so-called views. In [5], we have developed a more complex refinement language, which allows the user to formalize complete developments as refinement trees. The newly-introduced language supports branching (using CASL architectural specifications), going thus beyond what can be expressed using views, and further refinement of leaves of a refinement tree, as well as composition of refinements. In Hets, refinements written in this language can be parsed and statically analyzed, and their correctness and consistency can be checked using the calculi introduced in [5]. Moreover, the refinement trees can be visualised and inspected from within Hets.

Database Support Tradtionally, Hets analysis specification libraries in a local session that cannot be made permanent. Recently, database (sqlite or Postgresql) has been added. Thus, it is possible to write all information of specification libraries, as well as proofs that have been carried out, into a database, und use this in later sessions, as well as for other tools like Ontohub.org.

References

1. E. Astesiano, H.-J. Kreowski, and B. Krieg-Brückner. *Algebraic Foundations of Systems Specification.* Springer, 1999.
2. M. Bidoit and Peter D. Mosses. *CASL — the Common Algebraic Specification Language: User Manual.* Number 2900 in LNCS. Springer Verlag, 2004.
3. 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.
4. 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.
5. Mihai Codescu, Till Mossakowski, Donald Sannella, and Andrzej Tarlecki. Specification refinements: Calculi, tools, and applications. *Sci. Comput. Program.*, 144:1–49, 2017.
6. Razvan Diaconescu. Grothendieck Institutions. *Applied Cat. Struct.*, 10:383–402, 2002.
7. 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.
8. Joseph Goguen and Grigore Roşu. Institution morphisms. *Formal aspects of computing*, 13:274–307, 2002.
9. T. Mossakowski. Comorphism-based Grothendieck logics. In K. Diks and W. Rytter, editors, *MFCS 2002*, volume 2420 of *LNCS*, pages 593–604. Springer, 2002.
10. 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.
11. Till Mossakowski, Christian Maeder, and Klaus Lüttich. The heterogeneous tool set. In Orna Grumberg and Michael Huth, editors, *TACAS*, number 4424 in LNCS, pages 519–522. Springer, 2007.
12. Peter D. Mosses, editor. *CASL Reference Manual.* Number 2960 in LNCS. Springer Verlag, 2004.
13. D. Sannella and A. Tarlecki. Specifications in an arbitrary institution. *Information and Computation*, 76:165–210, 1988.
14. Donald Sannella and Andrzej Tarlecki. *Foundations of Algebraic Specification and Formal Software Development.* EATCS Monographs in Theoretical Computer Science. Springer, 2012.
15. 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.