ASMs and refinement of state-based systems

Gerhard Schellhorn

Institut für Informatik, Universität Augsburg,
D-86135 Augsburg, Germany

Abstract. My talk will present the refinement concept of abstract state machines (ASMs, [Gur95], [BS03]) that was introduced by Börger in the 90'ies ([Bör90a], [Bör90b]), when he found that traditional refinement concepts were not suitable to capture refinement in the verification of a Prolog compiler. His methodology for the use of the ASM refinement paradigm and numerous applications are described in [Bör03].

In the first half of my talk I will focus on the formalization of the refinement concept of ASMs, that I have done in [Sch01]. One of the main design goals of this formalization was effective deduction support, so verification conditions were expressed in Dynamic Logic. Dynamic Logic combines the imperative, operational style of specifying transitions as used in ASMs with predicate logic to specify properties. Since specification and verification using Dynamic Logic is supported by the KIV system [RSSB98], machine-supported correctness proofs for the theory could be done, and it could be successfully applied to formalize correctness proofs for the already mentioned Prolog compiler [SA98],[Sch99].

ASM refinement uses commuting diagrams to define the refinement relation, which are more general than those used in other refinement concepts. ASM refinement must also cope with the expressivity of ASM rules, which may be non-deterministic or fail to compute successor states (divergence) for certain states. ASM runs may be finite or infinite. The implied generality of the refinement concept allows to view ASM refinement as a framework which can be instantiated by other notions of refinement for state-based systems.

In the second half of my talk I will specifically consider how various forms of data refinement are related, as described in [Sch05]. I will present results relating the forward simulation proof technique of ASMs to classical data refinement [HHS86], to the behavioral approach [BDW99] and to the contract approach [WD96].

Recently a number of generalizations have been defined for data refinement. Two important ones are weak refinement ([DB01]) which divides operations into external and internal ones, and coupled refinement [DW03] where one operation can be implemented by a sequence of operations. I will show that both generalizations can be derived as instances of ASM refinement.

I will conclude the talk by discussing some of the open issues in the formalization of ASM refinement like a fully formal treatment of I/O or the relation to event-based formalisms such as CSP.

References

[BDW99] C. Bolton, J. Davies, and J.C.P. Woodcock. On the refinement and simulation of data types and processes. In K. Araki, A. Galloway, and K. Taguchi, editors, *Proceedings of the International conference of Integrated Formal Methods (IFM)*, pages 273–292. Springer, 1999.

[Bör90a] E. Börger. A Logical Operational Semantics for Full Prolog. Part I: Selection Core and Control. In E. Börger, H. Kleine Büning, M. M. Richter, and W. Schönfeld, editors, *CSL'89. 3rd Workshop on Computer Science Logic*, volume 440 of *LNCS*, pages 36–64. Springer, 1990.

[Bör90b] E. Börger. A Logical Operational Semantics of Full Prolog. Part II: Built-in Predicates for Database Manipulation. In B. Rovan, editor, *Mathematical Foundations of Computer Science*, volume 452 of *LNCS*, pages 1–14. Springer, 1990.

[Bör03] E. Börger. The ASM Refinement Method. *Formal Aspects of Computing*, 15 (1–2):237–257, November 2003.

[BS03] E. Börger and R. F. Stärk. *Abstract State Machines—A Method for High-Level System Design and Analysis.* Springer-Verlag, 2003.

[DB01] J. Derrick and E. Boiten. *Refinement in Z and in Object-Z : Foundations and Advanced Applications.* FACIT. Springer, 2001.

[DW03] J. Derrick and H. Wehrheim. Using Coupled Simulations in Non-atomic Refinement. In D. Bert, J. Bowen, S. King, and M. Walden, editors, *ZB 2003: Formal Specification and Development in Z and B*, volume 2651, pages 127–147. Springer LNCS, 2003.

[Gur95] M. Gurevich. Evolving algebras 1993: Lipari guide. In E. Börger, editor, *Specification and Validation Methods*, pages 9 – 36. Oxford University Press, 1995.

[HHS86] He Jifeng, C. A. R. Hoare, and J. W. Sanders. Data refinement refined. In B. Robinet and R. Wilhelm, editors, *Proc. ESOP 86*, volume 213 of *Lecture Notes in Computer Science*, pages 187–196. Springer-Verlag, 1986.

[RSSB98] W. Reif, G. Schellhorn, K. Stenzel, and M. Balser. Structured specifications and interactive proofs with KIV. In W. Bibel and P. Schmitt, editors, *Automated Deduction—A Basis for Applications*, pages 13 – 39. Kluwer Academic Publishers, Dordrecht, 1998.

[SA98] G. Schellhorn and W. Ahrendt. The WAM Case Study: Verifying Compiler Correctness for Prolog with KIV. In W. Bibel and P. Schmitt, editors, *Automated Deduction—A Basis for Applications*, pages 165 – 194. Kluwer Academic Publishers, Dordrecht, 1998.

[Sch99] G. Schellhorn. *Verification of Abstract State Machines.* PhD thesis, Universität Ulm, Fakultät für Informatik, 1999. (available at `www.informatik.uni-augsburg.de/swt/Publications.htm`).

[Sch01] G. Schellhorn. Verification of ASM Refinements Using Generalized Forward Simulation. *Journal of Universal Computer Science (J.UCS)*, 7(11):952–979, 2001. available at `http://hyperg.iicm.tu-graz.ac.at/jucs/`.

[Sch05] G. Schellhorn. ASM Refinement and Generalizations of Forward Simulation in Data Refinement: A Comparison. *Journal of Theoretical Computer Science*, 2005. accepted, to appear early this year.

[WD96] J. C. P. Woodcock and J. Davies. *Using Z: Specification, Proof and Refinement.* Prentice Hall International Series in Computer Science, 1996.