

Major Threat: From Formal Methods without Tools to Tools without Formal Methods

Bernhard Steffen
Chair of Programming Systems
University of Dortmund
44221 Dortmund, Germany
steffen@cs.uni-dortmund.de

Formal methods, typically conceptually based on algebraic approaches and various variants of mathematical logics, played a dominant role in the early days of software engineering. Originally their role was merely descriptive, and their intention was towards *complete* and *rigorous* specifications. Approaches like those based on VDM, Z, and B require a deep theoretical background, and their validation is deemed to be a laborious manual task.

In the late eighties a complementary approach started to grow, aiming at the development of *decidable* specification formalisms. These methods, based e.g. on type theory, model checking techniques, or abstract interpretation, were still fully formal, but thanks to their automation they were accessible to a much wider public. Their impact was and still is strongly linked to Moore's law, since they directly and tangibly profit from the progress in computing power and memory size and speed.

A third strand came in with the more recent advances of software engineering, notably the UML and the numerous Integrated Development Environments, which provide a wealth of loosely related features. This comprises various editors for graphical notations, analyzers for syntactic structures, generators of skeleton code from diagrams, debuggers, and simulators. More than 90% of today's code is written on the basis of this approach, often in a rather

experimental fashion. In fact, the success of this approach seems to be completely independent of all the foundational work of the two formal methods-based strands.

This development in software engineering seems quite different from the observations in programming language development, where the enormous impact of type theory is unquestioned. The panel:

Major Threat: From Formal Methods without Tools to Tools without Formal Methods

moderated by Bernhard Steffen (University of Dortmund) reviews this development, discusses the perspectives, and establishes recommendations for future directions of software engineering, and, in particular, for the future role of formal methods in this area. The panelists,

- Manfred Broy (Technical University of Munich),
- Mike Hinchey, (NASA Goddard Space Flight Center),
- Luigi Lavazza (Politecnico di Milano), and
- Tiziana Margaria (University of Göttingen),

cover a wide spectrum of knowledge, experience, and opinions - ranging from theory to practice and from the mathematical to the engineering perspective, thus representing all three strands.