Specifications in Software Development

Jeannette M. Wing
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA, 15213

Abstract

My tutorial presents through examples various kinds of specifications used during software development. It focuses on the practical aspects of the nature and use of formal specifications. I mention some open research problems that should be of particular interest to the LICS community.

1 Summary

I present through a set of examples various kinds of specifications used during software development. These specifications range in degree of formality. Informal ones such as dataflow diagrams [5] and structure charts [8] are most commonly used in practice today; rigorous ones such as VDM [4], Z [6], and Larch [3], and ones based on mechanized theorem proving such as Boyer-Moore logic [1] and higher-order logic (HOL) [2] are gaining visibility as concern for reliability of large, complex systems grows. The examples illustrate differences among specifications by *what* is being specified, e.g., algebras, programs, or computations, *how* they look, e.g., graphical or textual, *who* benefits from using them, e.g, end-users, designers, or programmers, and *when* in the software lifecycle they are most appropriately used, e.g., requirements analysis, detailed design, or validation and verification.

I cite non-trivial case studies done in university, commercial, and government sectors. I mention some open problems to show how far research has yet to go to meet practice. The recent establishment of standards committees advocating the use of formal methods for application areas such as safety-critical systems serves as a reminder to researchers that practitioners may expect more from formal methods than they can deliver. This tutorial draws on my introductory article on formal methods [7].

References

[1] R. S. Boyer and J S. Moore, *A Computational Logic*, Academic Press, New York, 1979.

[2] M. Gordon, "HOL: A Proof Generating System for Higher-Order Logic," *VLSI Specification, Verification and Synthesis*, G.V. Birtwistle and P.A. Subrahmanyam, editors, Kluwer, 1987.

[3] J.V. Guttag and J.J. Horning and J.M. Wing, "The Larch Family of Specification Languages," *IEEE Software*, Vol. 2, No. 5, September 1985, pp. 24-36.

[4] C.B. Jones, *Systematic Software Development Using VDM*, Prentice-Hall International, 1986.

[5] D.T. Ross, "Structured Analysis (SA): A Language for Communicating Ideas," *IEEE TSE*, January 1977, pp. 16-34.

[6] J.M. Spivey, *Introducing Z: a Specification Language and its Formal Semantics*, Cambridge University Press, 1988.

[7] J.M. Wing, "A Specifier's Introduction to Formal Methods," *IEEE Computer*, Vol. 23, No. 9, September 1990, pp. 8-24.

[8] E. Yourdon and L. Constantine, *Structured Design*, Prentice-Hall, Englewood Cliffs, 1979.

0-8186-2735-2/92 $3.00 © 1992 IEEE