Formal Methods: An International Perspective

Susan Gerhart
MCC Software Technology Program
3500 West Balcones Center Drive, Austin Texas 78759
gerhart@mcc.com 512-338-3492

Formal Methods has become a recognized body of knowledge over the past few years. Conference names, company and project titles, special publications, and government research programs all legitimize and define the area. Expressed as simply as possible, the goal of Formal Methods is to base the software development process squarely upon a workable set of mathematical techniques. The common names associated with various sub-classes of Formal Methods express both the purpose and mode of the technique: formal specification, mathematical verification, proofs of correctness, formal description languages, rigorous development methods, stepwise refinement, etc. There are many named methods --- VDM, Z, Larch, traces, functional verification, etc. -- and many associated with tools -- GYPSY, HOL, Boyer-Moore, B, etc.

Like some other technical subjects, e.g. artificial intelligence, Formal Methods brings in considerable intellectual history and a certain amount of controversy. What is a proof? How rigorously should engineering methods be applied? What is mathematically tractable? What does mathematics have to do with the physical and information worlds? Are we talking about the "maths" of Mathematics Departments, that of Engineering Schools, or something else? There's even a budding "sociology of mathematics".

The field of Formal Methods has taken on one other dimension. Influenced by funding patterns, cultural roots, and simply chance, North American and European research groups took different technical directions. Understand and reconciling the essential differences to improve the overall approach is not easy, and probably confuses practitioners and researchers outside the field. Hence, the purpose of this talk is to draw out some of the international forces from and upon the Formal Methods field:

1. Safety-critical (or high-integrity) systems vs. general software practice. Applicability and acceptance of the methods differs many ways, e.g. the cost-benefit tradeoffs and the influences of government regulators versus the marketplace. Formal methods for hardware design and verification are accelerating the demands for better software formal methods. Examples include: proofs of security properties, verification of floating point packages, formalization for reuse frameworks, and semantic definition during language design.

2. Verified system software vs. formally specified applications. Both European and North American research communities are addressing each objective, but tend to differ in assessing their merits. Verified system software seeks to assure that compiler down to hardware systems software is fully specified and verified. This well-defined problem differs from the objectives of specified applications in both the formalization challenge (compiler-assembler-circuit behavior being better formalized than information and reactive system properties) and the techniques applied (verification to assure the absence of certain kinds of errors vs. validation to assure adequate understanding of system purpose). The international differences lie primarily in where to place the emphasis of research and application and how to ameliorate the results of doing one to a much greater extent than the other. Progress has been faster when the objective is "merely" specification. An associated factor is the U.S. "requirement" for tools in contrast with the European self-confidence in intellectual skills. Examples include: the Computational Logic and ESPRIT ProCoS stacks (or towers), the claims of the "verified" VIPER microprocessor, and the Oxford-Hursley CICS projects.

CH2982-7/91/0000/0036$01.00 © 1991 IEEE

3. The effects of emerging and spreading standards employing formal methods. Standards serve many purposes: protective measures for citizenry and for business and national institutions, as well as stimulating science and industry. The U.K. MoD-0055 Standard is the prime example in both its characterization of the role of formal methods in assuring high-integrity systems and in the business and educational enterprise spawned to meet its far-reaching goals and spread its effects beyond military into financial, medical, and transportation systems.

4. Ramifications for the educational process. Mathematics is a particularly touchy subject for U.S. educators, making advocacy of mathematically-based methods even more problematic. Several problems must be faced: defining the kind of mathematics (discrete structures and logic) to be taught well throughout the education system, teaching the skills for formalizing problems and behaviors, and adjusting the level of rigor to fit software development processes. Examples will be given of what's being taught, e.g. in the U.K. universities and through the U.S. SEI, as well as the opportunities for stop-gap educational measures.

In summary, this talk provides the perspective of one U.S. researcher who has been particularly influenced by the international forces shaping the Formal Methods field as both a technical subject and a social enterprise.

References:

Formal Methods for Trustworthy Systems (FM89), Springer-Verlag Workshops in Computing, D. Craigen & K. Summerskill, Editors, ISBN-0-387-19635-8, 1990.

"Seven Myths of Formal Methods", Anthony Hall, IEEE Software, September 1990.