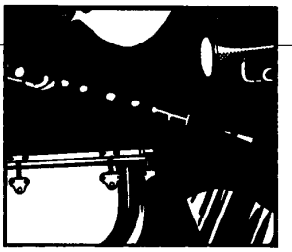
Author(s): Blyth, David; Boldyreff, Cornelia; Ruggles, Clive; Tetteh-Lartey, Nik.
Article title: The Case for Formal Methods in Standards
Year of publication: 1990
Citation: Blyth, D. et al. (1990) 'The Case for Formal Methods in Standards' IEEE Software 7 (5) 65 - 67
Link to published version: http://dx.doi.org/10.1109/52.57893
DOI: 10.1109/52.57893



The Case for Formal Methods in Standards

David Blyth, Cornelia Boldyreff, Clive Ruggles, and **Nik Tetteh-Lartey**
British Computer Society

Applying formal methods to standards making would result in more accurate, more understandable, and more useful standards.

Formal methods are useful for more than applications development. They can help standards developers create better standards and help ensure that the standards' content is correctly understood by those applying standards.

Recognizing the lack of informed opinion on the use of formal methods in standards development, production, and verification, the British Computer Society formed a working group to address the problem. This group has brought together people with a wide breadth of experience in the use of formal methods inside and outside the standards-making process. Their interests span areas like communication protocols, the specification of programming languages, graphics standardization, and document structure.

The BCS's efforts have underscored the value of formal methods in standards making, as well as deriving caveats and guidelines for formal methods' use in standards making.

The authors are all members of the British Computer Society's Working Group on Formal Methods in Standards. A fuller account of the group's work has been published in *Formal Methods in Standards: A Report from the BCS Working Group*, edited by Clive Ruggles (Springer-Verlag, 1990).

How formal methods can help

The general aim of those people developing standards is to ensure that a standard is useful, usable, compatible with existing standards, maintainable, and error-free. To meet these aims, standards makers should seriously consider using formal methods. Throughout the standards-development process, formal methods have a beneficial role to play.

The main potential benefit of using formal methods in a standard's development and expression is improving the standard's quality. In standards development as in engineering, "quality" means

0740-7459/90/0900/0065/$01.00 © 1990 IEEE

fitness for purpose.

In the early stages of standards development, formal methods can result in considerable clarification during the development and expression of the underlying conceptual model for a standard or family of related standards.

They can also define precisely the relation among the components of both the standard being developed and other standards. This aids integration by letting you formulate a set of standards in a compatible notation and by letting you assess as a whole the set's formal properties (like mutual consistency). In short, suitable formal methods could provide an excellent basis for project planning of the standards development within the International Standards Organization and other standards-making bodies.

Later in the development process, formal methods can improve a standard's quality during its use by letting it be expressed clearly, unambiguously, and concisely in a way that natural language, however carefully restricted, does not allow. An associated benefit of reducing the dependence on a particular natural language is improving communication of technical concepts among the people speaking different languages in the international standard-development process.

Finally, formal methods can aid standards development at the maintenance stage, by letting you, for example, formally prove the adequacy of a proposed change. The availability of tools like theorem provers should help considerably in reducing maintenance costs both to standards developers and users.

Formal methods can improve a standard's quality during its use by letting it be expressed clearly, unambiguously, and concisely.

The central issue in assessing whether to use formal methods in any particular standard is the importance of correctness (that no errors are introduced between specifying what the standard is about and the more detailed development of the standard itself). Without formal methods the odds are against correctness. Incorrectness is by far the most intractable fault in poor standards, in terms of both errors introduced and ambiguities and inadequacies in the specification upon which the standard is based. This argues strongly for the speedy introduction of formal methods into standards. With formal methods, you can prove that the specification has specific required properties, which helps identify inadequacies, and you can uncover ambiguities for rewriting as unambiguous expressions.

Formal methods consist of or incorporate a formal description technique; they may also provide the mathematical apparatus whereby you can check design steps for correctness with respect to the specification. In the development of both software and standards specifications, you may contrast formal description techniques with informal description techniques like those that rely on the use of natural languages.

Natural languages let you use idioms and imprecisely defined terms, which leads to ambiguities. A formal description technique is based on a symbolic notation (its metalanguage, known also as a formal specification language) that uses rigorous and unambiguous rules both for developing expressions in the language (its syn-

Where formal methods have been applied to standards

Formal methods have been used in several standards areas, although in varying degrees. Examples include the following.

• Programming languages. The application of formal methods in the development of language standards is not widespread. Until recently, no common terms of reference had emerged for the definition of languages. For example, the Modula-2 standard is now being defined formally using the Vienna Development Method specification language but the idea of a formal definition was rejected in the case of the ANSI X3.159 C standard. Recently, ISO guidelines have been drawn up and published as a technical report.[1]

• Office documents. Office Document Architecture is a multipart international standard (ISO 8613) driven by the need for the open transfer of office documents. It standardizes the semantics of the structural and content elements of documents. Although the ODA standard is not formally expressed, a formal specification of ODA (FODA) is being developed within the ISO/International Electronics Commission as an addendum to the standard.

• Graphics. The Graphical Kernel System became the first ISO graphics standard in 1985 (ISO 7942). The Programmer's Hierarchical Interactive Graphics System (ISO 9592) became an ISO standard in 1989. In addition to GKS and PHIGS, there are three other associated international graphics standards: Computer Graphics Metafile (ISO 8632), Computer Graphics Interface (ISO 9636), and GKS-3D (ISO 8805). While none of these standards is formally defined, many efforts were made in the late 1980s, particularly by David Duce and his collaborators,[3] to apply formal techniques in an attempt to clearly understand the concepts underlying GKS and PHIGS.

• Communications. The use of formal description techniques for the specification of ISO Open System Interconnection standards and Comité Consultatif International de Télégraphique et Téléphonique international telecommunications standards is well advanced. Three such formal description techniques used in this area have themselves been standardized: the Estelle extended finite-state-machine language, LOTOS temporal-ordering specification language, and Specification and Description Language.

References

1. B. Meek, "Language Standards Committees and Revisions," *SIGPlan Notices*, Dec. 1988, pp. 134-142.
2. D.A. Duce and M.S. Parsons, "GKS: Some Lessons Learnt from Formal Specifications," *Proc. GKS Review Workshop*, Eurographics Assn., Geneva, 1987.
3. D.A. Duce, P.J.W. ten Hagen, and R. van Liese, "Components, Frameworks, and GKS Input," *Proc. Eurographics 89 Conf.*, North-Holland, Amsterdam, 1989.

tax) and interpreting the meaning of these expressions (its semantics).

Although usually presented in contrast to each other, these two techniques — formal and informal — are best viewed as complementary. A formal specification may be accompanied by a natural-language commentary; or a natural-language specification may be supplemented by formal expressions of some of its parts. This is like knowing that something has been proved and using the result — if you need to look at the formal proof, it is available. Much engineering proceeds in this way.

Guidelines

The introduction of formal methods can be achieved only through education. Appreciating the need for a gradual migration toward a fuller use of formal methods, the International Standards Organization has recommended a three-phase plan[1] to introduce formal methods into standards:

- In phase 1, where the use of formal methods is restricted due to lack of expertise, their use should be encouraged as a parallel activity to formulating the standard in a natural language. Insights gained from the formalization may contribute to the quality of the standard by, for example, improving error detection. The plan recommends that any formalization work be published as a technical report to make this work accessible among ISO members.
- In phase 2, building on increased knowledge and experience in the use of formal methods, development of the formally expressed version of the standard should proceed in parallel with its natural-language version and be published as an informative annex to the standard.
- Once there is widespread knowledge of formal methods, in phase 3, standards should take the form of a formal description with a complementary natural-language description.

Ideally, the application of formal methods should be undertaken as an integral part of the standards-development process.

Avoid retroactive formalization. While retroactively applying formal methods is possible when an existing standard requires revision, perhaps updating and clarification, such retroactive application of formal methods can cause major problems.

This was true, for example, in the Graphical Kernel System standard.[2] In trying to specify parts of the standard formally, many deficiencies in the original natural language standard were uncovered, like insufficient abstraction and lack of hierarchical structure in the underlying data model, ambiguities, and confusing and misleading nomenclature. This forced the retroactive formalizers to make decisions to overcome these deficiencies and proceed with the formal definition.

Ideally, the application of formal methods should be undertaken as an integral part of the standards-development process.

Unfortunately, such decisions have little value unless successfully argued through the standards-review process. Quickly coordinating changes with review is essential. Reversing a single decision may lead to extensive changes, so it is desirable that the formal development does not proceed too far beyond the review process. But neither dare it lag too far behind, lest crucial issues fail to be identified before it is too late to consider them in the review.

In most cases, it might be more sensible simply to abandon a nonformal standard and start again from scratch. After all, exercises in retroactive formalization tend to reveal such a lack of conceptual integrity and clarity in a standard that the revised standard would bear little resemblance to the original.

Choosing the methods. The choice of appropriate formal methods is a key factor in their application. The choice should be guided by technical considerations rather than political factors like "not invented here" syndrome. Appropriate factors include adequacy for expressing the proposed standard's content, sufficiency of its underlying mathematical basis for intended applications, accessibility of its notational form to the community of experts framing the standard, and availability of supporting tools.

We cannot overemphasize the importance of tools to support the use of formal methods. Such tools take the form of editors, syntax and type checkers, animators, proof checkers, and transformation systems. Although there are many such tools from research projects, there is a dearth of production-quality tools. Fortunately, the situation is improving. Those tools that do exist help not only the standard developers but also industry users wanting to intercept the standards to gain familiarity with them through, for example, animation tools.

Furthermore, when a formally specified standard comes to be implemented, its implementation may be facilitated by a supporting transformation system.

There is no single formal method that serves all these purposes equally well for all applications. Closer collaboration is required between the standards and research communities as formal methods suitable for standards work continue to be developed. And these methods must themselves be standardized. ❖

Acknowledgment

In writing this article, we drew on discussions and material developed by the BCS Working Group on Formal Methods in Standards, whose work we gratefully acknowledge as our primary source.

References

1. "JTC1 Statement of Policy on Formal Description Techniques," ISO/IEC JTC1 N145 and ISO/IEC JTC1/SC18 N1333, Int'l Standards Org., Geneva, 1987.
2. C.L.N. Ruggles and S.T. Yee, "Clarification through Formal Specification: Some Notes on Attempting a Top-Down Formal Specification of GKS in Meta-IV," *Proc. GKS Review Workshop*, Eurographics Assn., Geneva, 1987, pp. 39-57.