Formal methods and their role in developing safe systems

Muffy Thomas
Department of Computing Science, University of Glasgow

March 20, 1995

Abstract

An IEE/BCS Workshop was held in January 1995 to consider the role of formal methods in developing safety-related systems. Points of consensus reached include: that formal methods provide just one of the many methods and tools for obtaining, and of demonstrating assurance of, the safety of systems software; that they are a means and not an end in themselves; that they may be the best means available for demonstrating some properties; that there are some system properties which are best captured informally, or may not be capable of assessment by formal methods; that formal methods should not preclude, but be applied in conjunction with other methods; and that it is desirable that national and international standards reflect this situation.

It was also recognised that safety-related systems have much in common with other classes of systems which must be reliable, particularly with respect to considerations of commercial pressures, the availability and adoption of standards, and availability and applicability of tools for formal methods. The need for government support for long term tool development and maintenance was identified, along with a set of guidelines for the use of formal methods in relation to safety-related systems.

Introduction

The Institute of Electrical Engineers and British Computer Society recently formed a Joint Working Party to consider the role of formal methods in developing safety-related systems. As part of this activity, a workshop was held to assess formal methods as applied to the design and implementation

of safety-related systems. The aim of the workshop was to gather together a number of people working in the field: from industry, academia, government agencies and regulatory bodies; to give them an opportunity to discuss their views on current practices, problems; and to see if a consensus of recommendations for the future role of formal methods in developing safe systems could be reached.

A total of 27 people participated in the workshop, mainly senior managers and academics, with 12 from industry, 8 from universities, 4 from government agencies/professional bodies (including the MOD, IEE, and BCS), and 3 from regulatory bodies (including the CAA and HSE). This report reflects upon some of the arguments put forward and the conclusions reached during the workshop, as well as incorporating the results of some subsequent discussions with participants.

By a *safety-related system* we mean one in which incorrect behaviour could lead to loss of human life, serious environmental damage, injuries to or illness of persons, or significant loss of life. A safety-related system must implement the safety functions necessary to achieve and maintain a safe state. Moreover, it should also achieve the necessary level (possibly with other safety-related systems) of safety integrity required for the implementation of the safety functions.

These kinds of systems are increasingly prevalent, across a number of sectors, including, for example, medical, transportation, and energy-related applications. Principal concerns for any such system are, for example,

- does it provide *all* the necessary functionality?
- does it fulfill the safety requirements?
- does it fulfill the reliability requirements?

The latter two concerns are normally qualified by acceptable rates of failure for the identified hazardous failure modes.

The record so far concerning safety-related computer systems, and of software in particular, has been very good (c.f. Professor Donald MacKenzie's recent study [1] which reported only 1,100 deaths, world-wide, in the last ten years from computer-related accidents – only 3 per cent of which can be strictly interpreted as resulting from software errors). However, technological advances and the increasing prevalence of these systems leave no

[1] Computer-related accidental death: an empirical exploration, *Science and Policy*, August 1994.

room for complacency. Moreover, particularly difficult problems for software, and ones which cause some concern within the community, are the issues associated with *certification* and *standards*.

One way to address these concerns is to use *formal methods*.

By *formal methods* we mean the use of mathematical notations and techniques, in particular those which support the development (i.e. specification, design, construction, and verification) and maintainance of computer-based systems, and particularly their software. The branch of mathematics of particular concern is discrete mathematics. Although continuous mathematics (e.g. laws of aerodynamics) is also relevant, particularly within the context of safety-related systems, it, and its application, is better understood and not of immediate concern here.

The workshop was focussed very much around the needs of safety-related systems, with formal methods as means to an end, rather than ends in themselves. The emphasis was on *properties*, and providing evidence for them. This is an important and useful focal concept, as it entails making explicit any assumptions about the context in which a safety-related system operates.

The Workshop considered questions such as:

- under what circumstances should formal methods be used?
- what should they be used for, and why?
- for what project stages should they be used?
- what are the disadvantages and limitations?
- what are the current good practices?
- what is the current position with regard to standards for formal methods and safety-related systems, and what should the position be?
- what further developments are needed?

The workshop was organised into three 'syndicates'; each syndicate meeting twice to discuss in depth two aspects of the role of formal methods in safety-related systems. After each syndicate meeting, the participants came together to discuss the outcomes of the syndicate discussions, and then again at the end of the workshop for a longer discussion on the issues which had been raised.

The remainder of the report is organised into the following five sections: formal methods, tools, standards, uptake, benefits to customer, and conclusions.

Formal methods

What are formal methods?

Formal methods entail more than the well-known specification, verification and refinement languages and methodologies such as VDM, Z, CSP, LOTOS, etc.; indeed the use of context–free grammars may be considered to be an early success of formal notations. What is crucial is that a formal notation, technique or method permits, indeed encourages, *reasoning*. By reasoning we mean both formal reasoning, about the specification/design/implementation, and also informal reasoning about the relationship between a formal model and a real world problem and requirements. The latter is just as important as the former.

Why use formal methods?

Formal notations offer *languages* in which to communicate, or express concepts and problems which are not well captured in other notations. A formal specification raises the potential for rigorous *refutation*.

Formal specifications and formal reasoning provide evidence, or assurance, for a safety case, which is usually more convincing way than informal evidence. This is a compelling reason to use formal methods, particularly when one is trying to satisfy legal constraints, or to demonstrate adherence to principles such as the HSE ALARP[2] principle. Moreover, cost and risk analysis techniques may be founded upon formal models of behaviour.

The issues of assurance and quantification of risk and cost aside, one of the most compelling reasons to use formal methods for safety-related systems, is that *formal methods entail making explicit assumptions about context*. This is particularly relevant where systems are embedded, as they most often are in safety-related systems. Formal methods not only encourage making assumptions explicit, but most formal reasoning techniques actually tease out any implicitly held assumptions, during the reasoning process.

[2] *as low as reasonably possible*

When and how to use formal methods

There are many dimensions to consider when evaluating when and how to use formal methods, for example,

- application domain (e.g. process control)
- domain for application of formal methods (e.g. software, hardware)
- personnel involved (e.g. experience, size of teams)
- project stage (requirements capture, design, testing)
- criticality of component.

It may be inappropriate to use formal methods because of inherent reasons such as timing constraints (of the operational behaviour of the system), or because a lack of time (for the software development), tools and mature techniques makes it impossible to apply them in a cost-effective manner. There are no hard and fast rules, rather judgements must be formed through experience.

For example, it is generally agreed that formal methods are appropriate when used by a small team of reasonably experienced professionals during the specification, modelling, and analysis stages of a critical system component, or novel algorithm; but they are inappropriate, for example, for use by a large inexperienced team considering requirements capture.

An important point to bear in mind is that formal methods can be used with *varying degrees of rigour*. For example, proof obligations need not be discharged, nor even stated; sometimes, merely using a formal notation in a specification or design is enough to gain major benefit.

The relationships between various formal methods and other methods, particularly testing, is not always clear; too often they are perceived as orthogonal concerns. However, formal methods have a lot to offer other methods: for example, black box test cases can be generated from formal specifications. Clearly more guidance on how to exploit relationships between particular formal methods and other methods, is needed.

There is some concern over the "presentability", or "readability" of formal notations and techniques. While education and better notations have enabled a wider range of professionals to use the techniques effectively, it must be recognised that what they produce will be "read" by readers with

different engineering, and other, backgrounds. Thus, there is a need to address the ways in which we communicate the results of formal analysis in a less formal way to colleagues and review bodies.

Guidelines for Use

During the workshop, the need for guidelines for use of formal methods – a *technical brief*, was clearly identified. The guidelines should cover when, why and how to use formal methods, and should be based upon case studies and experience. The guidelines would *not* (indeed cannot) be prescriptive, but rather would provide informed, mature guidance for developers of safety-related systems. More specifically, the guidelines would offer advice on:

- particular methods and notations, with case studies of typical problems best tackled with each method/notation,
- criteria for analysing when and how to use formal methods,
- the use of formal methods with respect to those criteria,
- relationships with other design and testing methods such as black box testing and structured walkthroughs,
- the use of formalism in presenting a safety-case.

Although it was not explicitly discussed during the workshop, it is recommended here that a detailed study should be carried out, under direction of the IEE/BCS working party, with the aim of producing such a set of guidelines.

Tools

Software tools supporting formal methods, e.g. from syntax and type checkers through to simulators, animators, prototypers, proof assistants and mechanised theorem provers are *essential* to the effective use of formal methods. Many good tools have already been developed, some of which have been used in projects to great effect. For example, there are refinement support tools such as the B-tool, CADiZ/ZETA; mechanised provers such as HOL, Proof Power, and LP (some of which have refinement applications built as front ends); rewriting tools such as RRL and MERILL; compilers, symbolic animators and model checkers for LOTOS and the Concurrency Workbench for

CCS (this is by no means an exhaustive list). While some tools are specifically devoted to the language and specific needs of a particular method, many tools are comparatively generic, and must be tailored to each specific use.

Because of diversity, both of formalisms and of applications, many tool users, and potential users, feel that current tools are too general. However, the safety-related systems market is very small, and it may not be reasonable to expect application area specific tools. This problem is shared, to some extent, with other sectors which use formal methods.

More pressing concerns are aspects of tool supply, namely *continuity* and *maintenance*. These are particular concerns in the context of safety-related systems with long lifetimes. A supplier who is developing a system which will run for, say, ten years, needs to know that the development tools used will also be supported over this period. Formal method tool support and maintenance is extremely expensive, and the long term viability of tool vendors, in particular, is an uncertainty. Whilst individual vendors may overcome some of the commercial pressures by selling consultancy as well as tools, the problems of financial commitment to long-term tool support remains. The market for formal methods tools is too small to leave the responsibility for this commitment to industry (i.e. to the manufacturer/supplier); government support is essential. Unfortunately, there is a lack of U.K. government policy and support in this regard, in contrast to Europe (e.g. France) and U.S.A. where there is considerably more government support.

Standards

Role of Standards

The role of a standard may be seen as that of an *arbiter* between the supplier of a system, the users of the system, and, ultimately, the public: the standard is a benchmark of best practice and reflects the current understanding of the responsibilities of the supplier and the user. The user is protected by, and gains benefit from, a standard whilst the supplier bears the cost of conformance to the standard. Their existence mean that both organisations and individuals can cite evidence of "best practice" by adhering to these standards.

While the primary aim of a standard is to lay down the "best practice" requirements for a product, process, or professional, secondary aims include establishing a common vocabulary and common tools, or tool requirements.

In general, standards also have serious commercial implications in relation to ensuring fair competition, and may open, or close, markets.

Another view of the role of the standard is that while on the one hand it defines a set of generic (perhaps industry-wide, or even international) requirements, on the other hand, it has to fulfill the role of the "ideal": a negotiated contract for that system.

All such aspects must be taken into account when balancing the tensions between desired and de facto standards.

Formal methods and standards

Standards for safety-related systems are clearly desirable, and several current standards require, or recommend, the use of formal methods during the development of software for some integrity levels. However, there is much criticism of these standards; for example the use of phrases such as "recommended" and "highly recommended" (c.f. IEC SC65A/1508) is very ambiguous. Also, the coupling of requirements for formal methods to integrity levels poses problems since the integrity levels of components typically change during the development of project.

Moreover, the pressures upon international standardisation bodies is such that in some cases, the emerging European and international standards are less demanding of formal methods than national standards. Such compromises are worrying, at least within the context of Europe, where the priority should be strengthening the requirements to demonstrate safety.

Rather than focus on the requirements (or lack) of formal methods as a problem, it may be more productive to concentrate on the *safety* properties and the need for evidence. That is, the purpose of standards is to establish the safety properties and to lay down the requirements for the derivation and presentation of evidence, or assurance, of the safety case. The emphasis should be on the *evidence*, rather than the method used to produce it. It follows then that formal methods will become more acceptable if and when software developers find such methods the *natural* means whereby evidence of adherence to a standard, in terms of the *safety* properties, is demonstrated.

Standards for Formal Methods

A further issue is that of standards for formal methods themselves. Several methods/techniques already have, or are in the process of having, interna-

tional (e.g. ISO) standards, e.g. LOTOS, SDL, Z, and VDM, to name a few. While the use of a standard inevitably means using "old" technology, standardised formal methods do ensure a common understanding of the methods used and may also encourage tool development. A compromise position is to require any extensions/deviations from a standard (for a formal method) to be defined and accepted by the relevant certification body, before they are used in a safety-case.

Uptake of formal methods

In common with other sectors using formal methods, the safety community perceives that the uptake of formal methods has not been as great as predicted a decade ago. Have they been both "over-sold and under-used" [3]?.

Tool functionality and availability, formal method presentation and education have been identified as barriers to acceptance, but they can, and are, being further developed. The question of market confidence remains: can the benefits of using formal methods be quantified? Perhaps the best reason for the use of formal methods lies in their effect on the development process as well as the final product, and in the economics of using them. In terms of the former, it is generally agreed that good use of formal methods often facilitates the understanding of a system and tends to bring about a related drive for simplicity. In terms of the latter, formal specification has been seen to deliver economies in terms of revealing errors at an important stage of the the development process; formal verification and refinement, on the other hand, have not generally been regarded as cost effective, except for the most critical of applications.

The demands of safety-related systems are different from other systems: they may be systems with long lifetimes, and increasingly, they must be developed according to stringent standards. The use of formal methods may soon be seen to be a cost effective way to meet these demands, particularly for applications which exhibit high integrity levels. However, the user, and potential user community must understand that formal methods should never preclude, but rather be applied in conjunction with other methods.

[3] Barroca and McDermid, Formal Methods: Use and Relevance for Safety-Critical Systems, *The Computer Journal* 35(6) 1992

Benefits to Customers

During the workshop, the Chairman repeatedly invited groups to consider the "customer" in relation to the perceived benefits of formal methods. The customer here may be seen as both the supplier/developer of a safety-related system, and the user of the system (either directly or indirectly) which may, in turn, be another supplier, or the public.

In this respect, formal methods may benefit the customer by providing the means for:

- precise communication and reasoning,
- evidence in a safety case,
- achieving economies of system development, and
- determining compliance to standards.

It should be stressed again that one of the most compelling reasons to use formal methods for *safety*-related systems, in particular, is that formal methods entail making assumptions about context explicit.

Conclusions

This report summarises some of the main points of consensus reached during the workshop. However, a single day's discussion, and subsequently this report, do not do justice to the complex nature of the topic under consideration. Rather, this is an appropriate starting point from which more detailed studies (e.g. on tool development and funding, guidelines for the use of formal methods in safety-related systems) should be undertaken. The IEE/BCW Working Party is encouraged to initiate and direct these studies.

The primary concern for safety-related systems is determining the *safety properties*; formal methods are a good means of providing evidence of such properties.

The safety-related community is a sophisticated and mature one seeking a serious and realistic role for formal methods. It acknowledges that there is no prescriptive way to use formal methods, and that they can be used with *varying* degrees of rigour; the use of a formal notation without formal analysis is often enough to gain major benefit.

Priorities identified by the workshop include:

- tools - increased support for development, continuity and maintenance, which implies the need for government support,
- standards which reflect the importance of *safety properties* and evidence of compliance,
- guidelines for the use of formal methods which include good case studies showing how formal methods can be employed, in conjuction with other methods, to provide evidence of safety properties.

Finally, formal methods will become more acceptable if and when software developers find such methods the *natural* means whereby evidence of *safety* properties, is demonstrated. This may well entail tackling the problem of effective, informal, communication of formal results.

Acknowledgements

The author acknowledges the contributions of the participants and speakers of the workshop, and offers particular thanks to Alasdair Kemp, John McDermid, and Andrew McGettrick for their helpful comments.

Comments and enquiries should be addressed to Alasdair Kemp, Institute of Electrical Engineers, Savoy Place, London WC2R OBL.